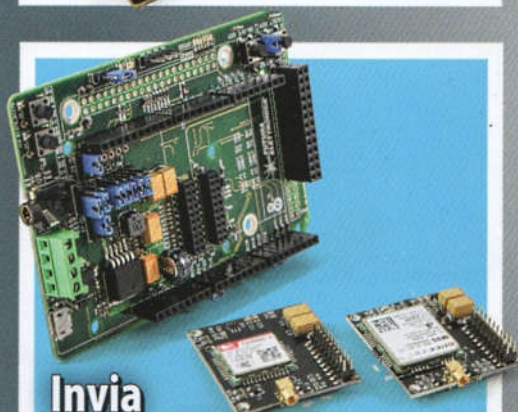
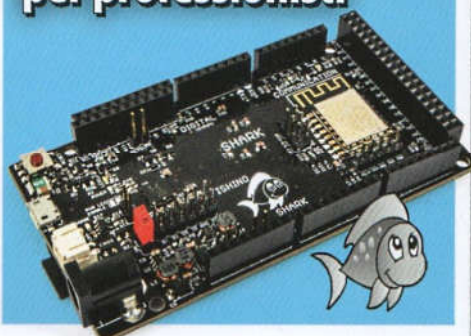


Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXIV - n. 229 - OTTOBRE 2018

SHARK:
la scheda a 32 bit
per professionisti



**Invia
SMS e chiamate con
Arduino e Raspberry Pi**

Impara a programmare con micro:bit



- Cooperative Scheduler: Arduino è multitasking
- Zerynth Studio: programma in Python

- Acchiappa la talpa: il gioco per tutti
- I sensori per l'Intelligenza Artificiale
- Bitcoin & Blockchain con Raspberry Pi
- Elettrometro con LED





cia

GROUP OF COMPANIES

Contact : sales@ciahk.net

VISIT US AT

Hall A1 booth 271

Electronica Munich 2018 Nov.13-16

cia
Printed Circuit Boards Ltd

Leading the way for 20 years in the supply and manufacture of professional printed circuit boards, from 1 to 36 Layers

Prototypes and **quick service** from our plant in Slovakia

Small, medium to large production from
4 principal JV manufacturing facilities in Mainland China



cia
Custom Products Ltd

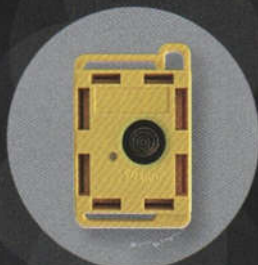
> **Focusing on the design, manufacture and distribution of custom made parts — principally:**

- Custom Made Connectors for Telecom and Automotive
- Rechargeable Battery Packs and Cells (NiMh, Li Ion, Li Polymer, Li FePO4 and Lead Acid)
- Primary Batteries (Li Coin, LiMnO2, LiSOCl2 & LiFeS2)
- Antenna products 433, 868, 1033 Mhz, Wifi, Bluetooth, GSM, GPS etc Turnkey products
- Antenna products, Ceramic Patch and chip antenna components
- Cable Assemblies and other Turnkey products



cia
Magconn™

Focusing on **the design, manufacture and distribution** of a innovative unique Magnetic Connection system using one part printed circuit board and one part connector



CIA : 1207 – 1210 Prosperity Place, 6 Shing Yip Street Kwun Tong Kowloon, Hong Kong, S A R C • www.ciahk.net



Direttore responsabile:

Arsenio Spadoni
(Arsenio.Spadoni@elettronica.in.it)

Redazione:

Stefano Garavaglia, Paolo Gaspari,
Boris Landoni, Davide Scullino,
Alessandro Sottocornola
(redazione@elettronica.in.it)

Grafica:

Alessia Sfulcini, Filippo Gori
(Alessia.Sfulcini@elettronica.in.it)

Ufficio Pubblicità:

Monica Premoli (0331-752668)
(Monica.Premoli@elettronica.in.it)

Ufficio Abbonamenti:

Elisa Guarniero (0331-752668)
(Elisa.Guarniero@elettronica.in.it)

DIREZIONE, REDAZIONE, PUBBLICITÀ:

FUTURA GROUP srl - Divisione Editoriale
via Adige 11 - 21013 Gallarate (VA)
Telefono 0331-752668 Fax 0331-792287

Abbonamenti:

Annuo 10 numeri Euro 45,00
Esteri 10 numeri Euro 45,00 (digitale)
Le richieste di abbonamento vanno inviate a:

FUTURA GROUP srl
via Adige 11, 21013 Gallarate (VA)
tel. 0331-752668.

Distribuzione per l'Italia:

SO.DI.P. Angelo Patuzzi S.p.A.
via Bettola 18 - 20092 Cinisello Balsamo (MI)
Telefono 02-660301 Fax 02-66030320

Stampa:

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

Elettronica In:

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.it/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.



© 2018 FUTURA GROUP srl

Gestire le ricadute del 5G

Il 5G è alle porte. Dopo la definizione degli standard, è iniziata la corsa di operatori e costruttori di terminali per dare vita alle prime reti commerciali che dovrebbero diventare operative entro la metà del 2019, mentre alcune reti sperimentali lo saranno già entro la fine di quest'anno.

Per le sue caratteristiche di velocità, latenza e banda, la tecnologia 5G sarà in grado di abilitare la nascita di una nuova generazione di applicazioni che avranno una ricaduta socio-economica di grande impatto, principalmente in cinque ambiti: salute, smart home/smart city, automotive, intrattenimento e industria.

Completare la rete mobile 5G in tutti i 28 Stati della Ue entro il 2020 costerà all'incirca 56 miliardi di euro, ma i benefici che ne deriveranno, sia in termini economici che occupazionali, saranno notevoli: il 5G dovrebbe infatti creare 2,3 milioni di posti di lavoro e contribuire a generare un beneficio annuo di 113,1 miliardi di euro, con ricadute sull'economia per 141 miliardi. Questo è quello che prospetta uno studio condotto dalla Commissione Europea con l'obiettivo di identificare e quantificare benefici, impatti e requisiti tecnici del 5G, e supportare così la pianificazione strategica per l'introduzione della tecnologia in tutta Europa. Secondo IDC, addirittura, a livello mondiale i servizi di mobility collegati al 5G rappresenteranno quasi il 60% della spesa nel periodo 2016-2021, con un valore che supererà i 1.000 miliardi di dollari nel 2021, grazie soprattutto agli investimenti delle aziende di telecomunicazioni. Insomma, tutti a tessere le lodi dei benefici e delle ricadute positive di questa tecnologia. È fuori di dubbio che il 5G porterà,

in generale, a un miglioramento delle condizioni di vita ma, come tutti i grandi cambiamenti, sarà sicuramente accompagnato da aspetti negativi che ad oggi non conosciamo. E qui non ci riferiamo alla maggior quantità di radiazioni che inevitabilmente saranno presenti nell'ambiente, bensì all'impatto sulla società e sui singoli individui.

Per spiegarci meglio: quando

si affermò Airbnb, le nuove opportunità di viaggiare e soggiornare in un modo più "smart" furono positivamente accolte da tutti. A distanza di anni, tuttavia, per effetto di

Airbnb i centri storici delle grandi città d'arte si sono svuotati, perdendo l'originale tessuto sociale, per diventare degli immensi quartieri dormitorio a beneficio dei clienti di questa società. Un effetto devastante che le Amministrazioni cittadine stanno ora cercando di contrastare. Nel caso del 5G, in ambito salute e benessere, ad esempio, prima o poi, saremo tutti muniti di dispositivi, connessi, in grado di misurare i parametri vitali: il tutto al fine di migliorare la qualità dell'assistenza medica e rendere più veloci le operazioni di soccorso. Tutto utile e corretto. Ma come cambierà la nostra vita, il nostro modo di relazionarci, la consapevolezza di noi stessi quando indosseremo questi dispositivi? E chi avrà accesso ai nostri dati? E come li utilizzerà? Bisogna essere consapevoli che tutte le grandi innovazioni tecnologiche presentano un aspetto negativo che bisogna essere pronti a gestire correttamente, se non vogliamo che gli svantaggi superino i benefici.



Arsenio Spadoni
(Arsenio.Spadoni@elettronica.in.it)

Sommario

Ottobre 2018
n. 229



IN COPERTINA:

Il simpatico robot che fa parte della Piattaforma Multifunzione costruita attorno alla scheda didattica micro:bit della BBC, un kit per chi comincia ed anche una board per sperimentare.

ELENCO INSERZIONISTI

- CIA
- Futura Elettronica
- Linear
- Maker Faire
- Microchip
- RM Elettronica

ARTICOLI

21 GADGET ACCHIAPPA LA TALPA

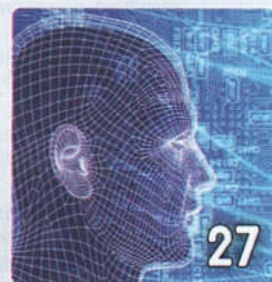
Simpatico passatempo di abilità, basato su LED e pulsanti che mette alla prova la vostra rapidità di reazione.



21

27 DIDATTICA SENSORI E INTELLIGENZA ARTIFICIALE

Facciamo una panoramica sulle soluzioni adottate nell'interazione tra robot e ambiente e integrate nel concetto di Machine Learning e in quello più ampio di AI.



27

33 APPLICAZIONI ZERYNTH STUDIO: PROGRAMMA I MICRO IN PYTHON

Ecco Zerynth Studio, la suite per applicazioni IoT che permette la programmazione in Python dei microcontrollori a 32 bit.



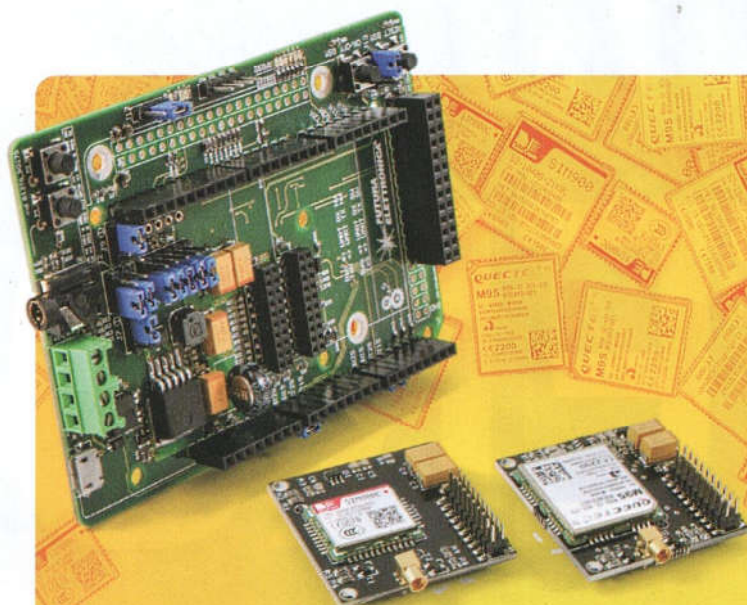
33

41 ARDUINO SHARK, A FISHINO SPUNTANO I DENTI

Una nuova scheda Arduino-like appare all'orizzonte ed ha prestazioni tali da poter "divorare" le concorrenti: non a caso è stata chiamata Shark, ossia squalo! Conosciamone le caratteristiche e le novità. Prima puntata.



41



55

ARDUINO GSM SHIELD UNIVERSALE

Supporta vari moduli GSM/GPRS low-cost allo scopo di dotare di connettività cellulare i nostri progetti con Arduino. Prima puntata.

69 ARDUINO COOPERATIVE SCHEDULER

Conosciamo un semplicissimo Scheduler Cooperativo in grado di gestire i task da far eseguire alle board Arduino e compatibili.



69

81 STRUMENTAZIONE ELETTROMETRO CON LED

Costruiamo un semplice e sensibile dispositivo per la misurazione dell'elettricità statica... e non solo.



81

La tiratura di questo numero è stata di 18.500 copie.



Mensile associato all'USPI,
Unione Stampa Periodica Italiana

91 DIDATTICA BITCOIN & BLOCKCHAIN

Approfittiamo di uno dei più noti esempi della vita reale per introdurre il concetto di blockchain, le sue caratteristiche e l'utilità in pratica, spiegando anche come minare i bitcoin utilizzando una piattaforma hardware essenziale come Raspberry Pi.



91

RUBRICHE

01 EDITORIALE

05 ELETTRONICA
INSIEME

12 APPUNTAMENTI
& EVENTI

17 COMPONENTI
& SISTEMI

110 SCIENZA
& TECNOLOGIA

123 EMBEDDED IOT

126 RINNOVABILI

99 DIDATTICA MICRO:BIT SMART CAR

Piattaforma multifunzione costruita attorno alla scheda didattica micro:bit della BBC. Integra un simpatico robot, un kit per chi comincia ed anche una board per sperimentare.



99

113 DIDATTICA MIT APP INVENTOR

Continuiamo il nostro viaggio alla scoperta di MIT App Inventor, un tool di sviluppo per applicazioni Android creato dal MIT. In questa sesta e conclusiva puntata analizzeremo le famiglie di componenti maps e connectivity e vedremo come pubblicare le nostre app su google play.



113



AHEAD OF WHAT'S POSSIBLE™

ZERO

LTC2063 1,4µA, ZERO-DRIFT OP AMP

- ▶ Basso consumo di corrente:
1,4µA tipico
- ▶ Offset di tensione:
5µV max
- ▶ Deriva termica:
0,02µV/°C max
- ▶ Corrente di Bias in ingresso:
30pA max, da -40°C a 85°C
100pA max, da -40°C a 125°C
- ▶ Filtro EMI integrato
(Reiezione di 114dB a 1,8GHz)
- ▶ 2mm x 2mm SC70

Offset 5µV • Drift 0,02µV/°C • Consumo 1,4µA • I_{BIAS} 0,5pA



Vieni a trovarci a



electronica 2018

Dal 13 al 16 Novembre a Monaco, Germania
Pad. C4, Stand 111



GUARDA IL VIDEO:
www.analog.com/LTC2063-video



CAMPIONI GRATUITI:
www.analog.com/LTC2063

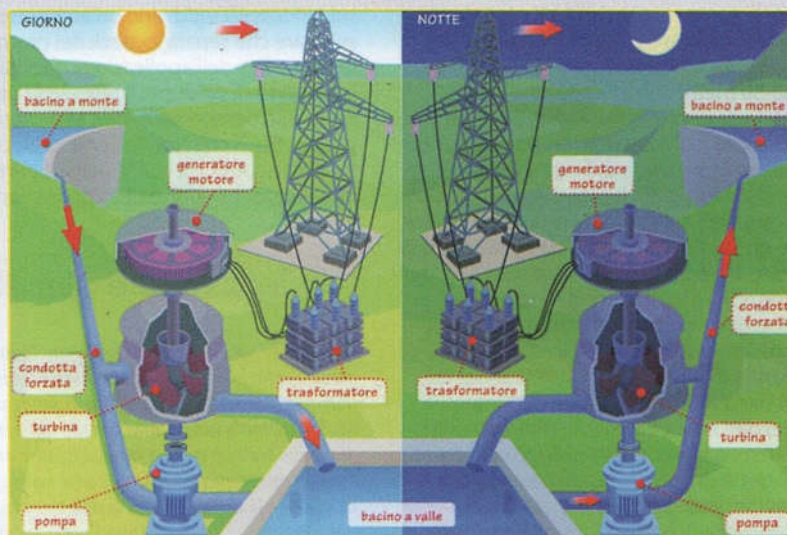
Queste pagine sono dedicate alle richieste, ai suggerimenti ed alle segnalazioni dei lettori. Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale. Contattateci numerosi all'indirizzo: redazione@elettronica.in.it

Accumulo idroelettrico, dove sta il business?

Recentemente mi è stata offerta la possibilità di investire in una società che possiede alcuni bacini idroelettrici con sistema di pompaggio e che offre tassi d'interesse di gran lunga superiori a quelli di mercato. La verità è che non capisco dove sta il business...

Marco Giuliani - Roma

I sistemi di accumulo idroelettrici sono basati sul pompaggio di acqua con due bacini posti a quote differenti: nelle ore in cui l'energia elettrica disponibile in rete è superiore all'energia richiesta, l'acqua viene pompata dal serbatoio inferiore al serbatoio superiore, mentre nelle ore di picco della domanda in cui c'è carenza di energia, l'acqua accumulata a monte viene rilasciata in apposite condotte e utilizzata per produrre energia elettrica, andando, alla fine, a riempire il bacino inferiore. Questi impianti sono essenziali per il funzionamento, la regolazione e la razionalizzazione delle reti di trasmissione dell'energia elettrica, soprattutto nei paesi in cui vengono utilizzate centrali termoelettriche o nucleari che non possono essere spente e riaccese rapidamente. Per quanto riguarda il ...business, il pompaggio entra in funzione quando c'è eccesso di energia, tipicamente di notte, e quando l'energia costa meno. In paesi come la Francia, con numerose centrali nucleari che non possono essere spente, il prezzo (all'ingrosso) dell'energia elettrica può scendere sotto 1 centesimo di euro per kWh; molti operatori, anche italiani, pagano



dunque cifre irrisorie per l'energia utilizzata per il pompaggio mentre ricavano da 5 a 10 centesimi al kWh quando, nelle ore di punta, immettono energia in rete facendo funzionare i loro impianti come generatori. Con in più un bonus, trattandosi di energia proveniente da fonte rinnovabile. Forse queste cifre possono spiegare l'elevata remunerazione che ti è stata offerta. Sta di fatto che attorno agli impianti di accumulo (anche di altro genere, batterie, supercondensatori, ecc...) c'è sempre più interesse da parte degli investitori professionali.

ANT+ un altro protocollo wireless per IoT

Oggi ho scoperto un nuovo protocollo wireless per IoT: ANT e ANT+, di cui però non sono riuscito a trovare alcuna informazione. Di cosa si tratta?

Luca Paciello - Napoli

Questo protocollo tanto nuovo non è, essendo stato rilasciato nel 2004 da Dynastream Innovations, una società canadese che ora fa parte di Garmin. ANT è paragonabile a Bluetooth Low Energy, essendo un protocollo wireless a bassissimo consumo che opera nella banda ISM dei 2,4 GHz. Come

Bluetooth Low Energy, è progettato per sensori alimentati da batterie a bottone della durata di mesi o anni. Dynastream Innovations non produce chip; dispone però di una gamma completa di moduli che operano con questo protocollo e che sono basati su chip di Nordic Semi-



conductor e Texas Instruments. Sebbene ANT sia un protocollo RF proprietario, l'interoperabilità è possibile attraverso la rete tramite ANT+. L'interoperabilità è assicurata dai profili dei dispositivi: qualsiasi dispositivo ANT+ che implementa uno specifico profilo è interoperabile con qualsiasi altro dispositivo ANT+ che ha lo stesso profilo. I nuovi prodotti devono superare un test di certificazione gestita dalla ANT+ Alliance. Originariamente ANT e ANT+ erano destinati al segmento sport e fitness, ma recentemente il prodotto è stato utilizzato per applicazioni nei settori

della domotica e dell'automazione industriale. Il protocollo viene sottoposto a sviluppo continuo e l'annuncio più recente riguarda il rilascio di ANT BLAZE, una tecnologia di rete per aziende mirata ad applicazioni IoT con un numero elevato di nodi.

Tra le caratteristiche di ANT+ la velocità di 20 kbit/s, la bassissima latenza, praticamente trascurabile, e la portata di circa 30-50 metri.

ANT usa uno schema TDMA (accesso multiplo a divisione di tempo) per suddividere ogni banda di frequenza di 1 MHz in slot temporali di circa 7 ms. I dispositivi accoppiati sul canale comunicano durante questi slot temporali, che si ripetono in base al periodo di messaggistica ANT (ad esempio, ogni 250 ms o 4 Hz). Nella pratica, decine o persino centinaia di nodi possono essere ospitati in una singola banda di frequenza da 1 MHz senza che vi siano interferenze, quando l'integrità dei dati è fondamentale, ANT può utilizzare una tecnica di messaggistica "burst". Si tratta di una tecnica di trasmissione multi-messaggio che utilizza l'intera larghezza di banda disponibile e viene eseguita fino al completamento della trasmissione di tutti i dati.

Alcuni dei canali RF ANT disponibili sono assegnati e regolati dalla ANT+ Alliance per mantenere l'integrità della rete e l'interoperabilità, ad esempio 2,450 e 2,457 GHz.

Per quanto riguarda l'hardware, attualmente sono disponibili le soluzioni on chip di TI (CC2570 e MSP430F2 MCU) e Nordic Semiconductor (SoC nRF52832) mentre Dynastream Innovations offre una soluzione modulare (D52 ANT). Le prime due opzioni richiedono componenti periferici aggiuntivi per formare un dispositivo funzionante, mentre il modulo è una soluzione hardware completamente testata.

Il protocollo ANT viene fornito come

stack di firmware autonomo, lasciando allo sviluppatore solo il compito di sviluppare il codice dell'applicazione. Lo sviluppo del codice applicativo per CC2570 viene eseguito utilizzando gli strumenti di sviluppo dell'MCU scelto, mentre il modulo nRF52832 SoC e D52 ANT (che si basa sul nRF52832) utilizzano un'architettura software che separa lo stack del protocollo dal codice dell'applicazione, facilitando il processo di sviluppo. Il nRF52832 utilizza un microprocessore ARM M4F incorporato. Sia TI che Nordic Semiconductor forniscono ANT DK e SDK per supportare lo sviluppo del codice dell'applicazione.

Vedere anche di notte

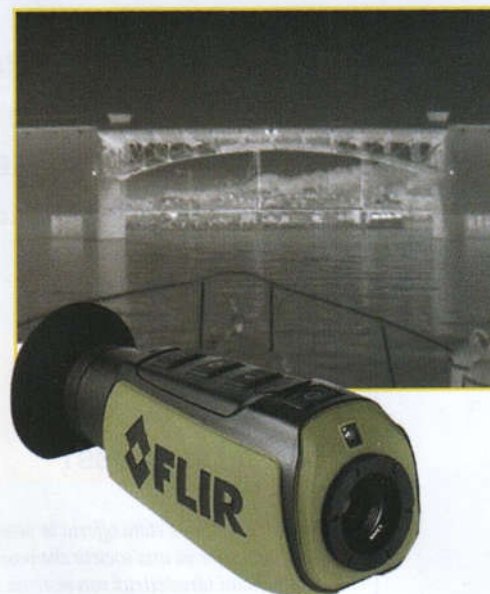
Navigando spesso anche di notte con la mia barca a vela, ho deciso di acquistare un visore notturno per rendere più sicura la navigazione. Ho tuttavia notato una grande differenza di prezzo tra i visori con intensificatore di luce e quelli con termocamera. Voi cosa mi suggerite? Qual è la differenza tra le due tecnologie?

Pino Fadda - Palermo

In effetti con 200 Euro puoi acquistare un buon visore notturno ad intensificatore di luce mentre ne devi spendere almeno 1.500 - 2.000 per acquistare un visore



Visore ad intensificatore di luce e con illuminatore IR Bushnell Equinox Z 6X50.



Visore notturno con termocamera FLIR Systems Scout II 240.

portatile con termocamera. Evidentemente per l'uso che devi fare, al di là dei pregi e dei difetti delle due tecnologie, la differenza di prezzo non giustifica l'acquisto di un modello con termocamera. Ma in cosa differiscono questi prodotti e come funzionano?

I sistemi ad intensificatore di luce amplificano la poca luce ambientale permettendo così di vedere anche in condizioni di luce molto scarsa; la visione si presenta così simile a quella che si presenterebbe di giorno. Questi dispositivi amplificano la luce esistente nell'ambiente, per cui, se l'ambiente fosse totalmente buio, non ci sarebbe nulla da amplificare e non si otterrebbe alcuna immagine. Ma il problema più grosso è quello opposto, ovvero quello dell'abbagliamento che si verifica quando nell'area inquadrata c'è una luce, anche relativamente poco intensa.



RM ELETTRONICA

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

Distributore autorizzato:

**FUTURA
ELETTRONICA**

**Il tuo punto
di riferimento
per l'elettronica
a ROMA**

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

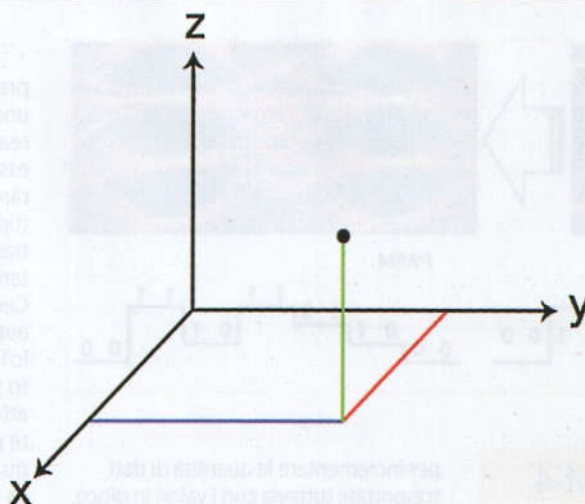
In questo caso si ha una sovraesposizione, con corone e scie luminose che ostacolano la visione.

Questi sistemi lavorano spesso in abbinamento all'infrarosso attivo, i loro sensori sono cioè sensibili alla radiazione infrarossa vicina che può essere di origine ambientale (se c'è ancora un po' di luce, per esempio al crepuscolo) o artificiale (cioè usando degli illuminatori all'infrarosso). Esistono due tipi principali di illuminatori all'infrarosso, quello classico, che opera sulla banda degli 850 nm e quello "nero" che emette sulla banda dei 940 nm; la differenza sta nel fatto che l'infrarosso classico produce una luce infrarossa non visibile ma se si guarda l'illuminatore si vedono i LED rossi, pur senza vedere tutta la luce che producono, mentre gli illuminatori con infrarosso "nero" a 940 nm producono luce infrarossa meno potente ma anche guardando gli illuminatori non si vede assolutamente nulla. Questi prodotti presentano un costo compreso tra 100 e 300 Euro, a seconda delle caratteristiche, dimensioni, ingrandimenti, ecc.

Un altro tipo di infrarosso è quello passivo, cioè quello termico: questo tipo di radiazione infrarossa è legata alla temperatura degli oggetti e degli esseri viventi e può essere individuata (e trasformata in immagine) da apposite telecamere dette termocamere. L'immagine prodotta dalle termocamere è completamente artificiale, viene infatti ricostruita totalmente dal sensore termico e dai chip della termocamera; generalmente viene usato il colore bianco per i punti a temperatura maggiore e il colore nero per i punti a temperatura minore, con tutta una serie di colori intermedi in scala di grigio per le temperature intermedie; è possibile dal menu della termocamera invertire questi colori, quindi il nero indicherà i punti più caldi e il bianco quelli più freddi oppure si possono usare delle LUT per i falsi colori, per esempio con i gialli per le temperature più calde e il blu per le temperature più fredde. Per poter vedere di notte, dunque, le termocamere non hanno necessità di luce ambiente o di luce artificiale.

Per quanto riguarda la tua applicazione, l'ambiente marittimo è spesso ingombro di luci brillanti sulla costa, e rappresenta uno scenario impegnativo per gli intensificatori; l'elevata luminosità in lontananza può abbagliare un visore notturno convenzionale, rendendo difficile l'osservazione di bersagli di interesse. E in assenza di luce, risultano del tutto inefficaci.

Al contrario, l'ambiente marittimo è favorevole alla tecnologia delle termocamere, poiché di solito si cerca un obiettivo caldo contro uno sfondo uniformemente freddo.



Localizzatore di precisione con Wi-Fi

È possibile localizzare con precisione uno smartphone (e il relativo utente) all'interno di edifici dove non è presente il segnale GPS utilizzando solamente la rete Wi-Fi?

Mario Deiana - Cagliari

Se per precisione intendiamo un valore dell'ordine del metro la risposta è no. Attualmente, in assenza di GPS, l'unica alternativa è l'impiego del Wi-Fi, ma la tecnologia disponibile non consente precisioni del genere. Tuttavia, così come per l'esterno, avere anche all'interno degli edifici un sistema standard ad elevata precisione consentirebbe di realizzare interessanti servizi basati sulla localizzazione. Basti pensare ad un ospedale, un edificio pubblico o un grande magazzino: il nostro smartphone ci potrebbe guidare verso l'ufficio, il laboratorio o il prodotto desiderato esattamente come il navigatore dell'automobile ci porta in quella tal via o piazza. L'elevata precisione potrebbe rendere possibile anche la gestione delle risorse, come la localizzazione di carrelli elevatori nei magazzini, la mancanza di sedie a rotelle negli aeroporti e lo spostamento di apparecchiature diagnostiche portatili negli ospedali. Un'altra applicazione è il geofencing, un perimetro virtuale che quando viene violato attiva un avviso. Avrebbe anche il potenziale

per competere con applicazioni come Google Nearby e altri approcci centrati sul beacon Bluetooth per l'uso in parcheggi pubblici, illuminazione stradale, telecamere di sicurezza e altri luoghi, nonché per un marketing mirato basato sulla posizione.

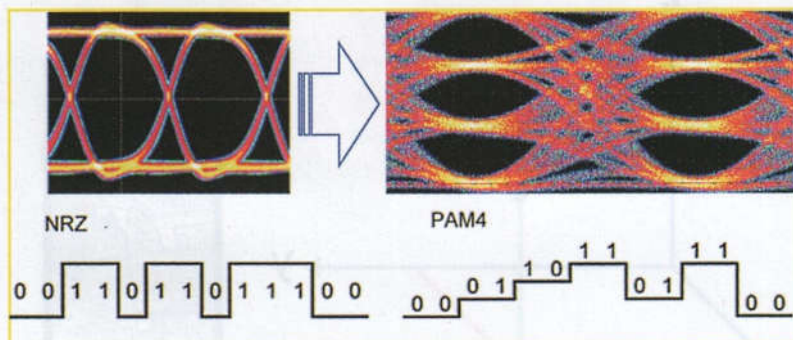
Fondamentale potrebbe risultare per le chiamate di emergenza, fornendo l'esatta posizione di chi ha effettuato la telefonata.

Per tutti questi (ottimi) motivi, da tempo le Associazioni preposte sono al lavoro, così come i produttori di smartphone. È stato così definito il protocollo IEEE 802.11mc Wi-Fi, conosciuto anche come Wi-Fi Round-Trip-Time (RTT) che permette di localizzare un utente anche se si trova al chiuso, grazie anche al nuovo sistema operativo Android P (e probabilmente anche ad un aggiornamento di iOS 11).

Wi-Fi RTT funziona secondo il protocollo FTC (Fine Timing Measurement) all'interno dello standard IEEE 802.11-2016 che utilizza una varietà di tecniche per individuare la posizione dello smartphone o del tablet. Sebbene 802.11mc sia stato definito da alcuni anni, è stato formalmente annunciato all'inizio del 2017 da Wi-Fi Alliance, che introduce anche la possibilità una "Location certificata Wi-Fi".

IEEE 802.11mc non è un semplice add-on ma un importante aggiornamento che richiede l'upgrade di tutti gli access point esistenti, con l'inclusione della esatta posizione, comprese le coordinate precise di latitudine, longitudine e altitudine. Un aggiornamento che potrebbe essere incentivato dalle nuove opportunità d'uso basate su questa tecnologia che sicuramente Google Maps e Apple Maps offriranno a breve. Inutile sottolineare come anche questa nuova tecnologia, per chi saprà immaginare applicazioni innovative, rappresenti una nuova possibilità di business per sviluppatori e start-up.





La codifica PAM-4

Quali sono le differenze tra le codifiche PAM-4 e NRZ utilizzate per il trasferimento dei dati digitali?

Marco Vitali – Milano

Per i fornitori di servizi Internet, trasferire un flusso di dati digitali in rete diventa sempre più complesso a causa della maggior velocità e quantità di dati richiesti dagli utenti. Ormai guardare un film o seguire un evento sportivo è più che normale per tutti noi; la cosa preoccupante per i fornitori di servizi è la somma dei dati richiesti che va ad intasare server, linee di connessioni coassiali o ottiche, backplane. In pochi anni le interfacce e le linee di questi sistemi si sono evolute passando da velocità di pochi Gigabaud a velocità dell'ordine di 200/400 GB. Per evitare di alzare ulteriormente la frequenza di clock si è pensato anche di utilizzare una diversa codifica del segnale. Attualmente viene utilizzato il sistema di codifica a 2 livelli NRZ (Non Return to Zero), mentre nei sistemi più sofisticati ha preso piede il formato PAM-4 (4 level Pulse Amplitude Modulation), o modulazione dell'ampiezza degli impulsi a 4 livelli. La modulazione PAM-4 prevede quattro valori ammessi per l'ampiezza di ciascun impulso, anziché i soli due della classica modulazione NRZ (o PAM-2). Con il segnale che è in grado di assumere 4 livelli, si possono codificare 2 bit. Teoricamente sarebbe possibile aumentare ancora il numero di livelli

per incrementare la quantità di dati trasportati; tuttavia con i valori in gioco, allo stato attuale la cosa diventa molto complessa e la possibilità di introdurre errori aumenta vertiginosamente. Per quanto riguarda l'analisi del segnale così codificato tramite oscilloscopio, il classico diagramma "ad occhio" si trasforma in un diagramma a tre occhi, rendendo più complesso lo studio della forma d'onda.

Gli smartphone per la sicurezza dei ponti

Dopo la tragedia di Genova, mi è venuto in mente che, forse, sfruttando gli accelerometri presenti nei nostri smartphone potremmo contribuire a valutare la solidità di ponti e viadotti in maniera relativamente semplice. Che ne dite?

Giuseppe Santo – Napoli

La tua non è un'idea bizzarra, tutt'altro! Basti pensare a come funziona Google Traffic, dove i dati della velocità di tutti gli smartphone presenti all'interno delle vetture che stanno viaggiando in una specifica strada consente ai server di Google di calcolare la

presenza di code o meno, offrendoci una situazione del traffico in tempo reale. La stessa tecnica potrebbe essere utilizzata per ricevere ed elaborare i dati provenienti da altri sensori (tipicamente l'accelerometro) quando transitiamo con la nostra vettura sui tanti ponti presenti nel nostro paese. Certo, installando appositi sensori autoalimentati e realizzando delle reti IoT specifiche, i dati ottenuti sarebbero sicuramente più validi; tuttavia, in attesa che ciò accada, la soluzione da te proposta potrebbe essere adottata quasi immediatamente.

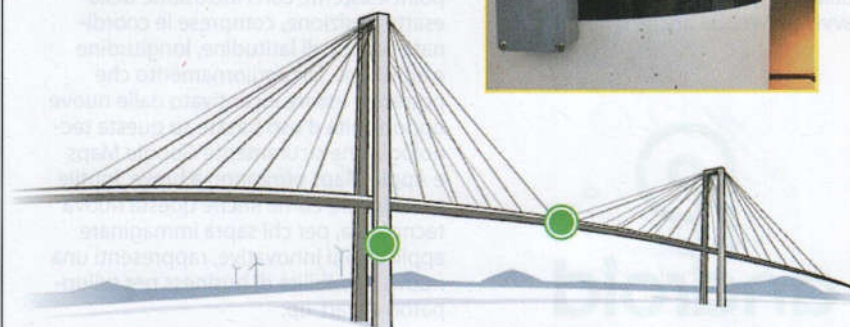
La riprova che la tua non è un'idea così bizzarra viene dal MIT (Massachusetts Institute of Technology) di Boston che ha condotto uno studio simile su un ponte di Boston, l'Harvard Bridge: i risultati sono stati molto incoraggianti, anche se sono stati necessari moltissimi dati per ricavare dati significativi. Quando la vettura e lo smartphone transitano su un ponte, l'accelerometro è in grado di captare le vibrazioni prodotte, e questo dato, insieme a quelli di tutti gli altri utenti, ha consentito ai ricercatori del MIT di desumere lo stato di salute strutturale del ponte. Si potrebbe quindi creare una rete di sensori mobili per il monitoraggio dei ponti, usando i dati raccolti dai telefonini degli automobilisti: una rete che sarebbe molto pervasiva ed economica.



Web Forum e supporto tecnico

Hai un problema con uno dei circuiti pubblicati? Vorresti effettuare una modifica? Accedi al nostro Web Forum dove i nostri tecnici (ma anche gli altri lettori) ti aiuteranno a chiarire qualsiasi dubbio di natura tecnica. Collegati a:

www.elettronica.in/webforum



La tensione giusta per ogni progetto!

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

CONVERTITORI STEPDOWN DC-DC

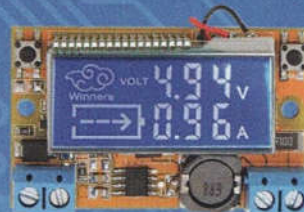
Da 5-36V a 1,25-32V - 5A



€ 14,00
cod. YB728

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

€ 9,90
cod. DCDISPAVDOWN



Da 7-28V a 5V - 3A



€ 2,50
cod. MP1584ENDCDC

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



€ 6,90
cod. STEPDOWNDISP

Da 9-35V
a 5V - 5A



€ 6,50
cod. DCDCSTEPDOWN

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



€ 5,90
cod. STEPD5V32

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



€ 4,30
cod. MODULODCDC

CONVERTITORI STEPUP DC-DC

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



€ 6,50
cod. STEPUP30V

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



€ 3,00
cod. STEPUPUSB

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

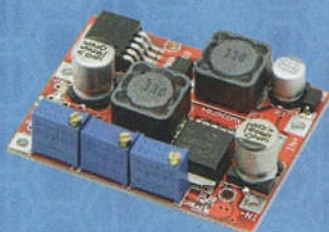


€ 4,90
cod. DCDC5V

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

CONVERTITORE STEPUP e STEPDOWN DC-DC

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



€ 16,00
cod. STEPUPDOWN



**FUTURA
ELETTRONICA®**
www.futurashop.it

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

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

Elettronica In TI GUIDA NEL FUTURO DELL'ELETTRONICA!

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 euro
45,00
anziché euro ~~60,00~~

COME ABBONARSI

① on-line

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

② mail

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

③ posta

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

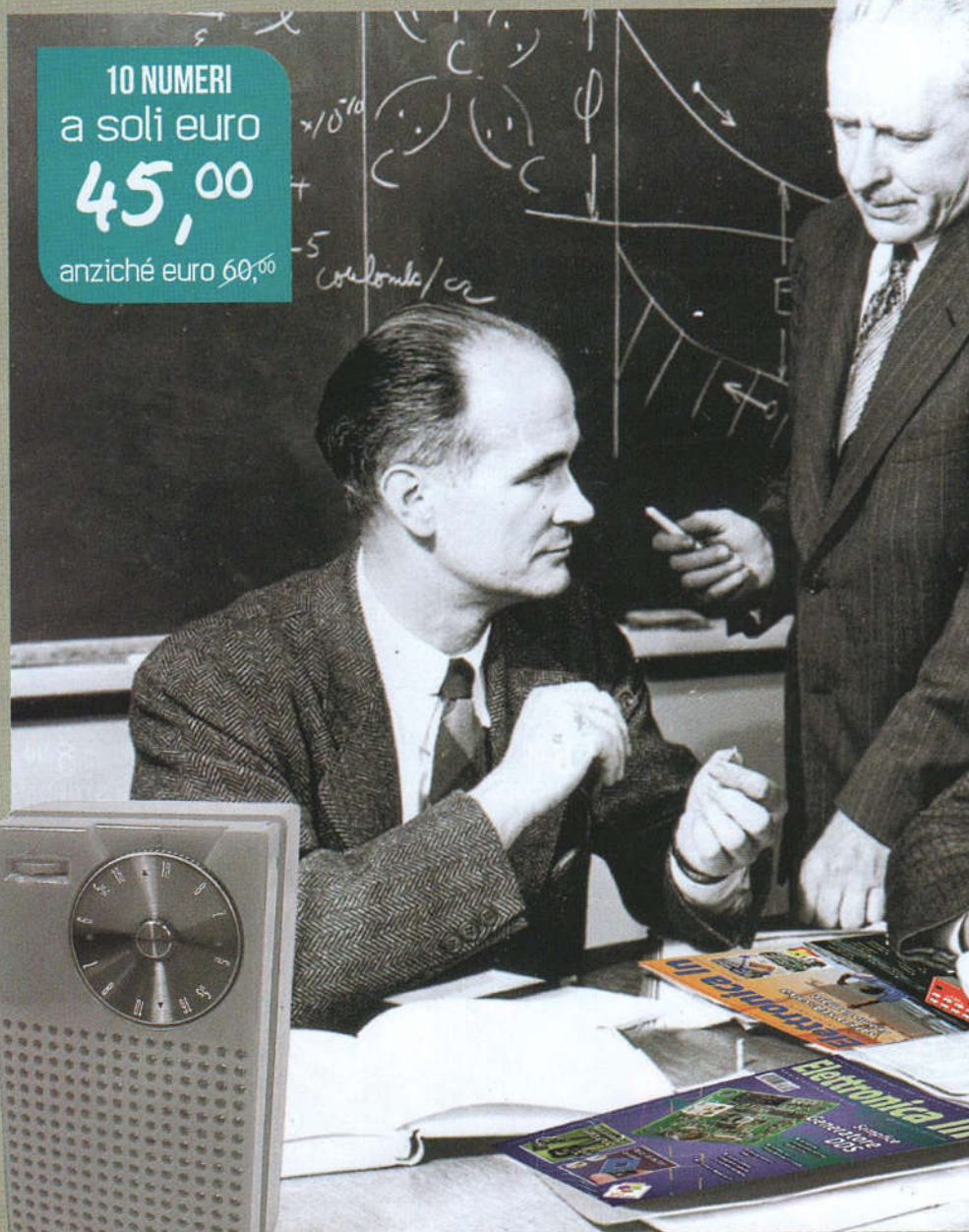
FUTURA GROUP srl
Via Adige 11
21013 Gallarate (VA)

④ telefono

telefonando a
+39-0331-752668

⑤ fax

compilando il modulo di abbonamento riportato a lato e inviandolo al seguente numero:
+39-0331-792287



PER I RINNOVI ATTENDERE NOSTRO AVVISO

UNO STRUMENTO INDISPENSABILE PER I TUOI STUDI, IL TUO LAVORO O LA TUA PASSIONE CHE PUOI RICEVERE COMODAMENTE A CASA TUA OGNI MESE SOTTOSCRIVENDO UN...

ABBONAMENTO ANNUALE



18 Ottobre 1954: viene lanciata sul mercato la prima radio a transistor portatile, la Regency TR1. D'ora in poi le ingombranti e fragilissime valvole verranno sostituite dal transistor inventato da Walter Brattain, William Shockley e John Bardeen, tre scienziati americani (nella foto qui a fianco) che nel 1947 realizzarono il primo prototipo funzionante. Ciò permetterà alla radio di "uscire di casa" e accompagnare la vita delle persone nelle strade, sulle macchine, nei parchi, sulle rive dei fiumi o sotto l'ombrellone.



IN OMAGGIO PER TE:



1 VOLUME A SCELTA

effettuati con la **DISCOUNT CARD** presso il punto vendita di Gallarate (VA) o nel nostro store on-line www.futurashop.it

della collana "L'Elettronica per tutti".

Sì

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



Tecnologia e apparati GPS



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 e degli omaggi.

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 • Fax: +39-0331-792287

Milano hi-fidelity 2018

Venticinquesima edizione milanese della manifestazione dedicata all'audio hi-end, LP, CD, musica liquida, mostra mercato nuovo e usato, concerti.

Milano hi-fidelity è una manifestazione dedicata agli appassionati di musica, di alta fedeltà, ai professionisti del settore, ai venditori, ed è promossa dalle case produttrici di alta fedeltà e dalle riviste del settore. L'obiettivo è



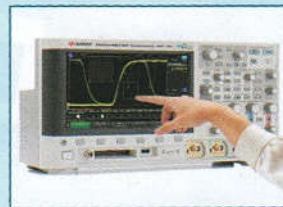
quello di offrire ai visitatori la possibilità di venire in contatto e di confrontarsi in diretta con gli operatori del settore, ascoltare le apparecchiature hi-end, fare esperienza delle tecnologie dell'audio di alta qualità e conoscere e ascoltare la musica "alla sorgente". La manifestazione si svolge sabato 6 e domenica 7 Ottobre 2018 presso l'Hotel Melià di Milano dalle ore 9:00 alle ore 18:00. Per tutta la giornata si svolgeranno demo e workshop in collaborazione con gli espositori. L'ingresso è completamente gratuito con possibilità di accedere a tutte le sale espositive e di partecipare a tutti gli eventi in programma.

<http://milano-roma-hi-fidelity-audio-show.it>

Fondamenti di Misura sull'Integrità dei Segnali

Il continuo aumento della quantità di dati da parte degli utenti, implica un sempre più vasto impiego di link digitali ad alta velocità nei prodotti elettronici, i quali aumentano di conseguenza la complessità dei progetti, anche dal punto di vista del test e debug. Avere l'opportunità di prendere in tempo le giuste decisioni, consente di evitare il re-spin dei prototipi e ridurre di conseguenza il time to market. Il seminario Keysight "Fondamenti di Misura sull'Integrità dei Segnali" affronta le sfide che i progettisti e gli sviluppatori di sistemi High Speed Digital devono affrontare a livello di componenti, circuiti e sistemi.

Cinque presentazioni molto focalizzate, forniranno una



visione approfondita sugli aspetti dell'integrità del segnale come Jitter, Crosstalk, TDR, power integrity, embedding & de-embedding, incluso l'utilizzo degli oscilloscopi nelle applicazioni RF.

Gli strumenti e le soluzioni Keysight saranno disponibili durante tutto il seminario per offrire l'opportunità di interagire uno-a-uno con esperti e prodotti Keysight. A Milano, il giorno 11 Ottobre 2018.

www.keysight.com

The Things Conference Italy

Zerynth, azienda leader negli strumenti di sviluppo IoT, organizza l'evento "The Things Conference Italy" che si terrà l'11 Ottobre 2018 presso il GATE Centre, un aggregatore di imprese al centro della Città della Galileiana di Pisa, a 5 minuti dalla Torre Pendente e dai principali laboratori universitari. La conferenza include talk di esperti provenienti da industria e accademia, che presenteranno lo stato dell'arte e le prospettive future della tecnologia wireless LoRaWAN, una delle tecnologie abilitanti l'Internet of Things e

l'Industry 4.0. Reti a ridotto consumo energetico con coperture di decine di Km, in grado di supportare progetti di Industrial Internet of Things e Industry 4.0 in settori diversi, come Smart City, agricoltura di precisione, gestione dei rifiuti e monitoraggio dei beni e delle risorse. L'evento sarà anche un'occasione unica per toccare con mano i tool di sviluppo messi a disposizione da aziende leader nel settore attraverso workshop e "hands on" organizzati da Zerynth, in collaborazione con The Things Network, Microchip Technology, RS Components, AAEON Europe, TOI - Things On Internet, Futura Elettronica e altri importanti player del settore. Questo il link per partecipare all'evento: <https://bit.ly/2PMxi5o>

www.zerynth.com



SAVE 2018

Il 17 e 18 ottobre è in programma a Veronafiere SAVE 2018, Mostra convegno delle Soluzioni e Applicazioni Verticali di Automazione, Strumentazione, Sensori; l'appuntamento verticale più atteso del settore torna a ottobre con un calendario colmo di appuntamenti, convegni, dibattiti, tavole rotonde, approfondimenti dal taglio tecnico-applicativo che affronteranno tematiche di grande attualità.

La due giorni veronese, organizzata da EIOM in collaborazione con le più importanti associazioni e istituzioni del settore, rappresenta da dodici edizioni una vetrina importante per aziende e operatori per avviare progetti con le eccellenze del settore e avere una panoramica completa sulle tendenze e le ultimissime novità per i differenti mercati industriali quali Energia, Ambiente, Food & Beverage, Pharma, Plastica, Chimica e Petrochimica, Automotive, Gomma e Plastica e così via.

Tanti gli argomenti sotto i riflettori: anche nel 2018 saranno prese in esame le innovazioni e soluzioni tecnologiche più attuali e importanti, con il tema della digitalizzazione ad impegnare trasversalmente molti fronti.

Un focus importante sarà dedicato alle tecnologie per l'alimentare, la visione e la tracciabilità: tematiche fondamentali per l'industria, esaminate in un convegno che toccherà un ventaglio di argomenti collegati alle necessità dell'industria alimentare, dall'efficienza energetica alla tracciabilità, fino alla logistica e alla sicurezza alimentare.

I maggiori player del mercato hanno già confermato la loro presenza all'evento, da anni SAVE è infatti un appuntamento fisso nell'agenda degli operatori, un momento fondamentale di aggiornamento professionale, di confronto diretto aperto a esperti, aziende, enti, istituzioni, per offrire la più completa rassegna di tecnologie e soluzioni nell'ambito della strumentazione e dell'automazione, oltre che quale punto d'incontro per sviluppare business.

Una ricca vetrina quindi quella prevista a SAVE il 17 e 18 ottobre, e per massimizzare le opportunità l'evento si svolgerà in concomitanza con MCM (Manutenzione Industriale e Asset Management), VPC (Valvole e Componenti), ACQUARIA (Trattamento acqua e aria), mcT Alimentare/ Visione e Tracciabilità (Food & Beverage, logistica e tracciabilità industriale), a comporre un grande evento sinergico ricco di soluzioni per tutti i professionisti che parteciperanno all'evento.

www.exposave.com

SAVE

17-18 ottobre 2018
Veronafiere

A ottobre a Bergamo il Festival della Scienza e la prima Bergamo Maker Faire

Dopo il successo dello scorso anno con 153.141 presenze, torna da sabato 6 a domenica 21 ottobre **BergamoScienza**, il festival di divulgazione scientifica organizzato dall'Associazione BergamoScienza. Anche quest'anno saranno 16 le giornate completamente dedicate alla scienza, in tutte le sue componenti. A Bergamo si daranno appuntamento scienziati di fama internazionale che in incontri, conferenze, dialoghi – tutti gratuiti – aggiungeranno come sempre con un linguaggio accessibile a tutti, l'ampio e variegato pubblico affezionato alla manifestazione sulle ultime scoperte scientifiche e tecnologiche.

BergamoScienza non è però solo conferenze: un nutrito programma di laboratori per scuole e ragazzi, fiore all'occhiello del festival, affiancherà mostre e spettacoli. Fin dalla sua nascita, BergamoScienza ha avuto come obiettivo prioritario il **coinvolgimento delle scuole**, per far appassionare i più giovani all'universo scientifico. Molte gli istituti che negli anni hanno partecipato al festival ideando e realizzando laboratori e veri e propri giochi scientifici, capaci di svelare il lato divertente e sorprendente della scienza e far scoprire il valore della collaborazione. Dalle 4 scuole presenti all'edizione del 2005 si è arrivati a 58 nel 2017 e quest'anno il

dato sale ancora: 64 scuole di Bergamo e provincia di ogni ordine e grado presenteranno 114 laboratori. Alcune di queste scuole – l'Istituto Comprensivo Solari di Albino, l'IC Trescore Balneario, l'ISIS Oscar Romero di Albino e l'ITIS Paleocapa – sono già state premiate con somme in denaro per l'originalità e la ricchezza di contenuti dei progetti presentati per la XVI edizione. Studenti e insegnanti saranno protagonisti della **Scuola in Piazza**, la consueta fiera scientifica on the road che anima il centro della città bassa trascinando passanti di tutte le età a provare esperimenti. BergamoScienza sarà anticipato quest'anno da una novità:

la prima **Maker Faire** di Bergamo. Da venerdì 28 a domenica

30 settembre al BergamoScienceCenter, studenti-inventori degli Istituti Tecnici e Professionali della Provincia mostreranno le loro innovative creazioni in campo scientifico e tecnologico.

www.bergamoscienza.it



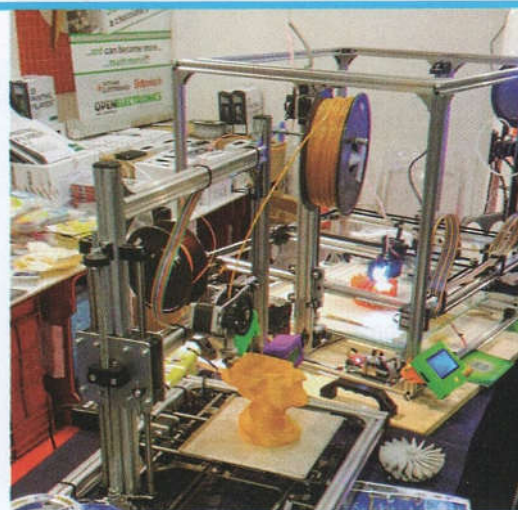
■ 12-14 OTTOBRE 2018

Maker Faire Rome – The European Edition

Fiera di Roma • Roma

Alla Maker Faire Rome si danno appuntamento gli appassionati di tecnologia, gli artigiani digitali, le scuole, le università, gli educatori, i centri di ricerca, gli artisti, gli studenti, le imprese e un pubblico di curiosi di ogni età pronti a fare un salto nel futuro. Per questa sesta edizione si prevedono numeri da record, se possibile superiori a quelli della passata edizione, sia per quanto riguarda i visitatori (che dovrebbero superare le 100 mila presenze), che per quanto riguarda i maker, le Scuole e le Università che hanno risposto alle relative Call con un numero impressionante di progetti. Solamente dalle Scuole sono arrivati

240 progetti innovativi che hanno messo in serio imbarazzo la Commissione tecnica incaricata di selezionare i migliori progetti, Commissione che ha individuato 55 progetti tra scuole superiori di secondo livello (Istituti tecnici e licei), ITS e scuole provenienti dalla Ue che li hanno elaborati: agli Istituti prescelti sarà garantito uno spazio gratuito alla Maker Faire e la possibilità di mettere in mostra, davanti a un vasto pubblico, la propria idea di futuro. Tanti, attuali e coinvolgenti i principali temi dell'edizione 2018: Industria 4.0, IoT ed elettronica, Intelligenza artificiale e Big Data, Smart Robotics e Smart Manufacturing, Mobilità intelligente, Education. Un intero padiglione sarà poi dedicato all'Agritech e al Foodtech dove saranno esposti progetti e iniziative legate a questi temi. Tra le tante novità dell'edizione 2018 c'è la Call for Space, una sezione speciale di Maker Faire Rome che offrirà a chiunque abbia sviluppato idee, prototipi e progetti nel settore spaziale di presentare le proprie realizzazioni al vasto pubblico della kermesse. Nell'anno in cui il progetto europeo Galileo di localizzazione satellitare è diventato operativo, con la Call for Space si offre l'oppor-



tunità unica di presentare le proprie realizzazioni e discuterle nel contesto dinamico e internazionale di Maker Faire Rome. Per celebrare questa novità assoluta l'evento, quest'anno, ospiterà una sezione celebrativa per collegare i makers "spaziali" di oggi con quelli di ieri. Il nucleo centrale di questa sezione sarà un'area riservata al tema dello spazio contenente una celebrazione dei makers del programma Apollo (di cui sta ricorrendo il cinquantennale che culminerà il prossimo luglio 2019 con l'anniversario del primo sbarco sulla Luna), e dei makers italiani che realizzarono i primi satelliti della serie San Marco. In quest'area verrà dato risalto al leggendario Apollo Guidance Computer, una delle più grandi innovazioni del programma Apollo, con la preziosa testimonianza di Don Eyles, programmatore e ingegnere del MIT, un maker d'eccezione che ha partecipato a tutte le missioni lunari. Maker Faire Rome - con una superficie espositiva di oltre 50.000 mq - è una grande festa dell'inventiva aperta ad ogni forma di creatività applicata: dalle idee più serie ed innovative fino alle più fantasiose, dalle nuove tecnologie al "saper fare" tradizionale ed artigianale rivisto in chiave 4.0.

Maker Faire



■ 23-24 OTTOBRE 2018

Smau 2018

Fieramilanocity • Milano

Condividere il percorso di innovazione è la chiave del successo, sia per le imprese che per le startup.

L'Open Innovation è, infatti, un meccanismo di contaminazione virtuoso dal quale non si può più prescindere per rimanere competitivi sul mercato: lo dimostra il Roadshow 2018 di Smau con le esperienze di numerose realtà tra cui Cisco, Intesa Sanpaolo, QVC, Seeweb,

TIM e altri. E Smau si conferma come il luogo ideale dove promuovere e discutere di Open Innovation.

A Smau Milano, dal 23 al 25 ottobre a Fieramilanocity, questo sarà il tema centrale di molti eventi.

L'Open Innovation, afferma Pierantonio Macola, Presidente Smau, è «un'opportunità non solo per le grandi imprese, ma anche per le medie e piccole realtà di qualsiasi settore produttivo, anche quelli più tradizionali, come l'agrifood e il manifatturiero, che possono contare su nuovi strumenti di lavoro in grado di garan-

tire efficienza e competitività, e su partner esterni, le startup innovative, che possono supportarle nel processo di Trasformazione Digitale. Smau, quindi, è l'occasione di trovare riunita in un unico evento nazionale e internazionale l'innovazione specializzata dei territori e i suoi protagonisti».

L'Open Innovation sarà fra i temi cardine di Smau Milano, con una serie di eventi dedicati: la presentazione, appunto, del rapporto 2018 dell'Osservatorio Open Innovation, il Premio Innovazione Smau (la call scade il 21 settembre), il Premio Lamarck per le startup più pronte



Area Kids: i giovanissimi, bambini e ragazzi dai 4 ai 15 anni, hanno un'area espositiva a loro dedicata, in cui possono sperimentare in prima persona il vero spirito maker, con laboratori di elettronica, robotica, programmazione e creatività digitale. Parola d'ordine "imparare facendo"!

Maker for Space: uno "spazio" per parlare di spazio, un'area dedicata alla celebrazione del Programma Apollo, un'anteprima assoluta del cinquantennale del primo sbarco sulla Luna che si celebrerà nel 2019.

Agrifood e Agritech: la sensoristica e le nuove tecnologie digitali applicate all'agricoltura e al cibo. Gli show case sul food e altre sorprese.

Negozio 4.0: come potenziare il punto vendita tradizionale con le nuove tecnologie digitali 4.0 ... ma non solo quelle, senza snaturarlo.

Robotics-AI Area: dopo il grande exploit del 2017, che ha visto la presenza delle migliori realtà italiane, quest'anno l'area della robotica cresce di importanza e si arricchisce di set up dimostrativi provenienti da tutta Europa.

Economia Circolare: Sostenibilità, Riciclo, Bioeconomy and Green Energy – coinvolgenti giochi a tema, laboratori e dimostrazioni pratiche, dibattiti e workshop, con l'obiettivo di sensibilizzare ed informare il grande pubblico. Insomma, un'immersione nell'innovazione che rende il mondo migliore. La parte formativa non è meno interessante: centinaia tra conferenze, seminari e workshop su temi come industria 4.0, domotica, droni e robotica, stampa 3D, IoT – Internet delle cose, tecnologia applicata ai mezzi di trasporto, cucina e musica. Da non perdere i demo-workshop, ovvero, i corsi dimostrativi sulle applicazioni della fabbricazione digitale.

Come sempre alle scuole è dedicato l'Educational Day, la visita in anteprima gratuita alla manifestazione che si svolge venerdì 12 ottobre tra le 9 e le 13.

<https://2018.makerfairerome.eu/it/>

a incontrare il mercato (in collaborazione con il Gruppo Giovani Imprenditori di Confindustria).

Poi ancora gli Smau Live Show, gli Startup Safari e l'evento ItaliaRestartsUp in collaborazione con Istituto del Commercio Estero e il Ministero dello Sviluppo Economico.

Attraverso il racconto di queste esperienze di innovazione Smau si propone di alimentare lo scambio e la contaminazione tra imprese, aziende digital, PA e startup. Per l'evento di Milano, il 23, 24 e 25 ottobre saranno selezionati 100 casi di successo di realtà provenienti da tutta Italia che saranno poi premiati nell'arco dei tre giorni.

www.smau.it

MOSTRE MERCATO

PADOVA

TUTTINFIERA 2018

Quartiere Fieristico - Padova
Organizzazione: Eboot Srl
Tel: 3776777342
eboot srl@gmail.com - www.eboot.it
03 e 04 Novembre 2018

CODEVILLA (PV)

FIERA DELL'ELETTRONICA

Expo Tucano SP1 N° 38 Codevilla
Organizzazione Eventi e Fiere
info@eventiefiere.com
www.eventiefiere.com
03 e 04 Novembre 2018

BASTIA UMBRA (PG)

EXPO ELETTRONICA

Quartiere Fieristico
Organizzazione: Blu Nautilus srl
Tel: 054153295 - Fax: 0541439584
info@blunautilus.it
www.expoelettronica.it
03 e 04 Novembre 2018

BRESCIA

FIERA DELL'ELETTRONICA

Polo fieristico Brescia
Organizzazione: Blu Nautilus srl
Tel: 054153295 - Fax: 0541439584
info@blunautilus.it
www.expoelettronica.it
10 e 11 Novembre 2018

VERONA

ELETTRO EXPO

Quartiere Fieristico di Verona
Organizzazione: Veronafiere
Tel: 0458298111 - Fax: 0458298288
info@veronafiere.it
www.veronafiere.it
10 e 11 Novembre 2018

PORDENONE

RADIOAMATORE2

Quartiere Espositivo - Pordenone
Organizzazione: Pordenone Fiere
Tel: 0434232111 Fax: 0434570415
info@fierapordenone.it
www.radioamatorepordenone.it
17 e 18 Novembre 2018

LUCCA

FIERA DI ELETTRONICA + LUCCA NERD

Polo Fiere - Via della Chiesa XXXII 237
Organizzazione: Prometeo
Tel: 057122266 - Fax: 05710831142
info@prometeo.tv
www.prometeo.tv
17 e 18 Novembre 2018

PESCARA

FIERA MERCATO NAZIONALE DEL RADIOAMATORE

Centro IBISCO - Città S. Angelo (PE)
Organizzazione: Ari Pescara
Tel: 0854215840
arifiera@aripescara.org
www.aripescara.org
24 e 25 Novembre 2018

BOLOGNA

EXPO ELETTRONICA

Bologna Fiere - Via Aldo Moro
Organizzazione: Blu Nautilus Srl
Tel: 0541439573 - Fax: 0541439584
info@expoelettronica.it
www.expoelettronica.it
24 e 25 Novembre 2018

ERBA (CO)

FIERA DELL'ELETTRONICA

Lario Fiere - Erba (CO)
Organizzazione: Lariofiere
Tel: 031637414
erbaelettronica@lariofiere.com
www.elettronicaerba.com
24 e 25 Novembre 2018

PARMA

FIERA DELL'ELETTRONICA & RADIOAMATORE

Wopa Temporary
via Palermo 6 - Parma
Organizzazione: Eboot Srl
Tel: 3776777342
eboot srl@gmail.com - www.eboot.it
24 e 25 Novembre 2018

L'elenco aggiornato di tutte le Mostre Mercato del 2018 è disponibile sul sito www.elettronica.in sul quale è anche possibile scrivere un commento sulle fiere visitate.

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

Futura Energy

Fai il pieno di energia con le batterie alcaline
di **FUTURA ELETTRONICA!**

**BATTERIE ALCALINE
LR6 AA - 1,5V**

Confezione da 24 pezzi



€ 7,50

Cod. 24LR6AA

**OGGI ANCORA PIÙ
CONVENIENTI NELLE
CONFEZIONI
DA 24 PEZZI!**

**BATTERIE ALCALINE
LR3 AAA - 1,5V**

Confezione da 24 pezzi



€ 6,50

Cod. 24LR03AAA

Prezzi IVA inclusa.



**BATTERIE
ALCALINE
LR6 AA 1,5V**

**Blister
da 4 pezzi**

€ 1,50

Cod. LR6-AA-4



**BATTERIE
ALCALINE
LR3 AAA - 1,5V**

**Blister
da 4 pezzi**

€ 1,50

Cod. LR03-AAA-4



**BATTERIA
ALCALINA
6LR61 - 9V**

**Blister
da 1 pezzo**

€ 1,80

Cod. 6LR61-9V-1



**BATTERIE
ALCALINE
23A - 12V**

**Blister
da 5 pezzi**

€ 2,50

Cod. 12V-23A-5



**BATTERIE
ALCALINE
LR14 C - 1,5V**

**Blister
da 2 pezzi**

€ 2,30

Cod. LR14-C-2



**BATTERIE
ALCALINE
LR20 D - 1,5V**

**Blister
da 2 pezzi**

€ 3,20

Cod. LR20-D-2

**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

ISL68300 e ISL68301, controllori DC/DC compatibili PMBus

Renesas Electronics annuncia la disponibilità di due nuovi controllori DC/DC, compatibili con il bus standard PMBus che forniscono una singola uscita di conversione Point Of Load (POL) adatti per alimentare FPGA, DSP, ASIC, Network Processors e sistemi di alimentazione general purpose. Il dispositivo ISL68300, che integra i drivers per MOSFET, ed il dispositivo ISL68301, con uscite PWM, semplificano la progettazione di alimentatori per data center, alimentatori per sistemi di comunicazione cablati e per sistemi di comunicazioni wireless così come la progettazione di alimentatori per



sistemi di automazione industriale. Il dispositivo ISL68300 è in grado di pilotare direttamente i MOSFET esterni mentre il dispositivo ISL68301 è stato progettato per funzionare in coppia con gli Smart Power Stage di Renesas o DrMOS power stage, sempre forniti

da Renesas, allo scopo di fornire una soluzione completa per la regolazione della tensione.

I controllori digitali ISL68300 e ISL68301 sono caratterizzati da un bus ad alta velocità per la condivisione delle correnti che gestisce fino a otto controllori per ottenere un sistema ad otto fasi a condivisione della corrente con valori totali che possono superare i 240 Ampere. Entrambi i controllori sono stati sviluppati utilizzando la tecnologia di controllo digitale della modulazione, proprietaria di Renesas, denominata ChargeMode.

www.renesas.com

AD7768-1, convertitore A/D sigma-delta per la conversione di precisione



Analog Devices ha presentato l'AD7768-1, un convertitore A/D sigma-delta ($\Sigma\Delta$) a 24 bit a canale singolo, a basso consumo e alte prestazioni per la conversione analogico-digitale di precisione di segnali in corrente continua e alternata. L'AD7768-1 consente di realizzare soluzioni a basso consumo per la strumentazione nella manutenzione predittiva (vibrazioni e fattore di potenza), test audio, acustica, verifiche di

integrità strutturale e apparecchi modulari per test automatici. Sono inoltre supportate applicazioni in campo medicale, come per l'Elettroencefalografia, l'Elettromiografia e l'Elettrocardiologia (EEG/EMG/ECG).

L'AD7768-1 offre una soluzione unica, configurabile e riutilizzabile per l'acquisizione di dati, stabilendo un nuovo standard in termini di prestazioni AC e DC combinate.

Questo consente ai progettisti di strumentazione e sistemi industriali di riutilizzare la catena di acquisizione principale, supportando più varianti di sistema, sia in applicazioni con isolamento che non isolate.

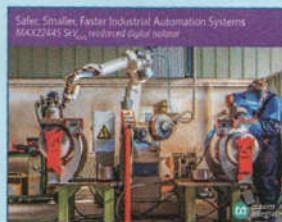
www.analog.com

MAX22445, isolatore digitale rinforzato da 5 kVrms

Per proteggere al meglio i sistemi industriali dai pericoli dei disturbi ad alta tensione, i progettisti di apparecchiature possono ora utilizzare il nuovo e affidabile isolatore digitale rinforzato a quattro canali MAX22445 da 5 kVRMS di Maxim Integrated. Con la capacità di offrire una maggiore velocità di circa 2 volte e con un consumo energetico inferiore di 4 volte rispetto alle soluzioni concorrenti, il MAX22445 fornisce comunicazioni affidabili attraverso la barriera di isolamento per garantire un funzionamento sicuro delle apparecchiature industriali, mediche e di altro tipo. Superando le carenze di altri isolatori digitali, il MAX22445 offre una

barriera rinforzata a doppio isolamento per segnali digitali veloci. L'IC trasferisce segnali fino a 200 Mbps impiegando una potenza significativamente inferiore rispetto alle soluzioni concorrenti. L'integrato è disponibile in un package SOIC wide-body a 16 pin con 8 mm di distanza di isolamento misurata sulla superficie (creepage) e in aria (clearance).

www.maximintegrated.com



Azionamento per motori microstepping ad alta definizione

Toshiba Electronics Europe ha presentato un nuovo azionamento per motori bipolari passo-passo con prestazioni da 50V/5A che supporta il funzionamento a micropassi 1/128. Il nuovo TB67S128FTG soddisfa i requisiti spesso in conflitto di alta velocità, prestazioni elevate e consumi ridotti, cosa che lo rende ideale per una varietà di applicazioni moderne, tra cui stampanti 3D, attrezzature per ufficio, distributori automatici di

banconote, macchine da gioco ed elettrodomestici. Al cuore del nuovo dispositivo si trova la tecnologia di controllo attivo del guadagno (AGC) di Toshiba che ottimizza la



corrente in base alla coppia richiesta, impedendo così lo stallo e consentendo di ottenere risparmi energetici. Il TB67S128FTG, molto diffuso nelle applicazioni d'ufficio dove l'elevata accuratezza e il funzionamento silenzioso sono critici, riduce il rumore e le vibrazioni fornendo 128 micropassi, rispetto ai 32 passi di altri IC. Esso incorpora inoltre un ritardo misto selezionabile (SMD) – una funzione di regolazione della corrente che

seleziona la migliore configurazione SMD e ottimizza la corrente di azionamento in base al motore. La tecnologia originale di rilevamento della corrente di Toshiba, ACDS (Advanced Current Detection System), permette il controllo della corrente del motore senza la necessità di un resistore di rilevamento di corrente esterno, riducendo lo spazio richiesto per il layout e i costi per la distinta materiali.

www.toshiba.semicon-storage.com

NI e Spirent Communications collaborano ad una soluzione sul test 5G



National Instruments e Spirent Communications hanno annunciato una collaborazione per la progettazione di sistemi di test per dispositivi 5G New Radio (NR). A dicembre 2017 il consorzio 3GPP ha rilasciato ufficialmente la prima bozza dello standard 5G NR, definendone una serie di caratteristiche. Lo standard subirà ulteriori mo-

difiche ed espansioni nei mesi e anni a venire. I sistemi di test devono perciò essere flessibili per potersi adattare agli aggiornamenti futuri e allo stesso tempo evitare di dover aggiornare l'hardware in base all'evoluzione dello standard. A questo scopo, Spirent Communications ha adottato i prodotti SDR (software defined radio) di NI per la crea-

zione delle proprie soluzioni di test per il 5G. Il sistema di Spirent impiega i dispositivi USRP (Universal Software Radio Peripheral) e il sistema transceiver a onde millimetriche di NI per la creazione di test che simulano le prestazioni del 5G NR durante la trasmissione di video, audio, dati e chiamate vocali.
ni.com/5g

LTM4678, regolatore μ Module da 50A scalabile fino a 250A

Analog Devices annuncia l'LTM4678, un regolatore step-down μ Module doppio da 25A o singolo da 50A con interfaccia digitale PMBus. L'LTM4678 utilizza due induttori esterni posti sopra il package BGA per dissipare calore dall'interno, mantenendo il dispositivo a bassa temperatura. La memoria EEPROM e l'interfaccia I2C/PMBus consentono all'utente di misurare, modificare e registrare i parametri di alimentazione principali come tensione, corrente di carico, temperatura e sequencing. Cinque regolatori LTM4678 possono ripartire la corrente a 50A ciascuno per fornire fino a 250A verso carichi come processori, FPGA e ASIC. Le applicazioni includono schede PCIe, sistemi di comunicazione, cloud computing, dispositivi ottici, medicali e industriali o strumentazione. L'LTM4678 integra un controllore DC/DC, una EEPROM, FET di potenza, induttori e componenti di supporto in un package BGA da 16mm x 16mm x 5,86mm. L'accuratezza della tensione di



uscita pari a $\pm 0,5\%$ è garantita per variazioni di linea, carico e temperatura nell'intervallo da -40°C a 125°C . L'LTM4678 può operare con un intervallo di tensioni di ingresso compreso fra 4,5V e 16V, e le due tensioni di uscita sono controllate digitalmente da 0,5V a 3,3V. L'LTM4678 raggiunge un livello di efficienza con picchi fino al 90% con 12VIN e 0,9VOUT a 50A, e fornisce 40A a 12VIN e 0,9VOUT con 70°C di temperatura ambiente e un flusso d'aria di raffreddamento a 200LFM. La frequenza di switching va da 350kHz a 1MHz e può essere sincronizzata a un clock esterno da 350kHz fino a 1MHz nel caso di applicazioni sensibili al rumore.

www.analog.com

Nuovi IGBT da 650V con tecnologia TRENCHSTOP IGBT6

Infineon Technologies introduce la tecnologia TRENCHSTOP IGBT6 di nuova generazione. Con una tensione di blocco da 650V, il dispositivo discreto è ottimizzato per applicazioni specifiche che richiedono lunga durata, alta affidabilità ed efficienza. Si tratta in particolare di elettrodomestici grandi e piccoli, macchine da cucire industriali e azionamenti per uso generale utilizzati in ventilatori e pompe che impiegano motori BLDC.

L'unione delle tecnologie trench e field-stop e l'impiego di un diodo di recupero in anti-parallelo Rapid 1, consente di ottenere perdite ridotte garantendo nel contempo nei sistemi di azionamento motori fino a 1 kW

buone prestazioni termiche, in particolare a frequenze di commutazione più elevate, che migliorano l'affidabilità e le tolleranze di progettazione. Le caratteristiche principali del TRENCHSTOP IGBT6 da 650 V sono una VCE di saturazione molto bassa ed

una elevata velocità di protezione da cortocircuito, migliore di 3 μsec . Il dispositivo è ottimizzato per la commutazione di frequenze comprese tra 5 kHz e 30 kHz ed è adatto per applicazioni che neces-



sitano di controllare in modo efficiente le emissioni parassite EMI. Attualmente sono disponibili le versioni a 8 e 15 ampere con package TO-220FP; altri package saranno introdotti nel 2019.

www.infineon.com

MCP39F511A, dual-mode Power Monitoring IC



Microchip Technology annuncia un flessibile IC per power monitoring dual-mode che misura sia in modalità AC che DC con la precisione più elevata del settore, pari ad un errore dello 0,1% su un ampio intervallo. Il calcolo dei dati di potenza e il monitoraggio degli eventi sono inclusi in un singolo IC, riducendo i costi di distinta base ed il tempo di svi-

luppo del firmware. L'IC per power monitoring MCP39F511A è un dispositivo altamente integrato che risponde alla crescente esigenza di misurazione più accurata di dati di potenza in progetti di elevate prestazioni. Il monitoraggio dell'alimentazione è diventato più diffuso nei mercati in crescita come Smart City, e Smart Home, e al contempo gli

sviluppatori cercano di monitorare le prestazioni del prodotto e migliorarne l'utilizzo dell'energia. Per semplificare ulteriormente gli sforzi di sviluppo, l'MCP39F511A include funzionalità avanzate come auto-save e auto-load, garantendo così che i risultati di misurazione non vengano mai persi.

www.microchip.com

Kit di ricarica wireless plug-and-play

Il kit di sviluppo del caricabatterie wireless plug-and-play di STMicroelectronics (STEWAL-15B045V1) consente di realizzare rapidamente caricabatterie ultra-compatti fino a 2,5 W con una bobina di 20 mm di diametro in grado di ricaricare piccoli sistemi IoT e dispositivi indossabili come orologi smart, attrezzature sportive o attrezzature sanitarie. Costruito attorno al controller del TX wireless di carica STWBC-WA, il kit comprende un'unità base di ricarica contenente un circuito trasmettitore con la bobina da 20 mm già collegata e pronta per l'uso. Utilizzando il software ST-SW-STWBCGUI basato su PC risulta particolarmente semplice configurare il circuito e monitorare le informazioni di runtime come l'energia erogata, la frequenza operativa,



la qualità della demodulazione e lo status del protocollo. Il kit include un dongle per l'esecuzione del software GUI. Il chip del controller STWBC-WA contiene driver integrati e supporta nativamente le topologie full-bridge o half-bridge per l'alimentazione dell'antenna. L'opzione half-bridge consente di caricare fino a 1W con una bobina di diametro inferiore, per un fattore di forma ancora più compatto. Il chip supporta tutte le funzioni standard di ricarica wireless, tra cui il rilevamento di oggetti estranei (FOD) e il rilevamento di presenza attiva per la ricarica sicura, utilizzando la retroazione digitale per adattare la potenza trasmessa e garantire un'efficienza ottimale in tutte le condizioni di carico.

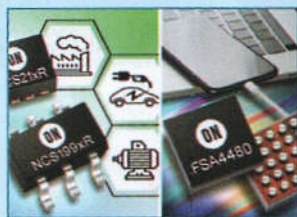
www.st.com

Da ON Semiconductor FSA4480 e NCS21x

ON Semiconductor ha annunciato l'introduzione di nuovi dispositivi che possono essere usati congiuntamente in applicazioni USB-C (Type-C), dispositivi che garantiscono notevoli vantaggi in numerose altre applicazioni dove i consumi rappresentano un elemento critico. Si tratta del chip FSA4480, uno switch audio analogico USB Type-C con funzioni di protezione integrate e della serie di amplificatori per il rilevamento della corrente NCS21x. FSA4480 è uno switch multimediale ad alte prestazioni per porte USB Type-C che supporta auricolari audio analogici. Questo dispositivo consente a una porta USB-Type-C comune di trasferire segnali USB 2.0, segnali audio analogici e segnali provenienti da un microfono analogico. Il nuovo FSA4480, in grado di supportare un percorso di rilevamento audio, si propone come una soluzione ottimizzata ad alto grado di

integrazione che permette di semplificare la fase di integrazione e minimizzare l'ingombro della soluzione complessiva. La famiglia NCS21x, formata da amplificatori per il rilevamento della corrente di tipo bi-direzionale che integrano resistori di precisione a elevato grado di adattamento, rappresenta una soluzione compatta e accurata per la misura di tensioni differenziali di valore ridotto. Questi dispositivi, in grado di rilevare tensioni di modo comune nell'intervallo compreso tra -0,3 e 26 V, sono ideali per il rilevamento della corrente "high side" e "low side".

www.onsemi.com



Omron amplia la gamma di relè di potenza AC bistabili

Omron Electronic Components Europe ha ampliato la gamma di relè di potenza AC bistabili ad alte prestazioni con nuove opzioni di connettività rivolte a una più ampia gamma di applicazioni tra cui stazioni di ricarica per veicoli elettrici, inverter per impianti fotovoltaici, controlli negli impianti di illuminazione e contatori intelligenti. I relè Omron G9TA e G9TB rappresentano una soluzione eccezionale per commutare correnti rispettivamente fino a 60A e 120A. Il relè di potenza AC bistabile Omron G9TA è progettato per assicurare commutazione ad alta potenza, bassa resistenza di contatto e resistenza alle vibrazioni per applicazioni fino a una corrente massima commutabile di 60A. Si ca-



atterizza per una resistenza di contatto di 2mΩ, capacità di sopportare impulsi di tensione fino a 6kV, tensione di commutazione di 250VAC. Per commutare correnti più alte, fino a 120A, è disponibile il relè di potenza AC bistabile Omron G9TB. Il relè utilizza contatti in lega d'argento; si caratterizza inoltre per una resistenza di contatto di 0,4mΩ, tensione di commutazione massima pari a 276VAC e una gamma di temperatura operativa compresa tra -40°C e 85°C.

<http://components.omron.eu>

Nuovi controller DLP Pico per scanner e stampanti 3D

Texas Instruments (TI) ha annunciato oggi nuovi controller DLP Pico per il mercato consumer che offrono funzionalità avanzate di controllo della luce per applicazioni in scanner e stampanti 3D, grazie anche al package di ridotte dimensioni. I controller DLPC347x offrono la risoluzione micron-sub-millimetrica tipica delle applicazioni ad alte prestazioni di livello industriale in un fattore di forma ridotto per le stampanti 3D desktop e gli scanner 3D portatili. Gli sviluppatori possono abbinare i nuovi controller DLPC3470, DLPC3478 o DLPC3479 ad uno dei quattro dispositivi DLP Pico con microspecchi (DMD) esistenti per realizzare una varietà di progetti portatili alimentati a batteria. L'alta risoluzione e l'elevata velocità di commutazione dei DMD e dei nuovi controller consentono velocità di stampa 3D

DLP® technology brings incredible accuracy to portable 3D scanning and 3D printing



cinque volte superiore rispetto alle tecnologie esistenti. Grazie al preciso controllo del pattern in un fattore di forma ridotto, gli sviluppatori possono catturare i dettagli di un oggetto con elevata precisione, a livello di micron, a vantaggio di applicazioni come la scansione dentale, la modellazione 3D e la visione 3D per la robotica. I campioni di pre-produzione dei controller DLPC3470, DLPC3478 e DLPC3479 sono già disponibili; la produzione in volumi è prevista per il 3Q 2018 per i modelli DLPC3470 e DLPC3478 e per il 1Q 2019 per il controller DLPC3479.

www.ti.com

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



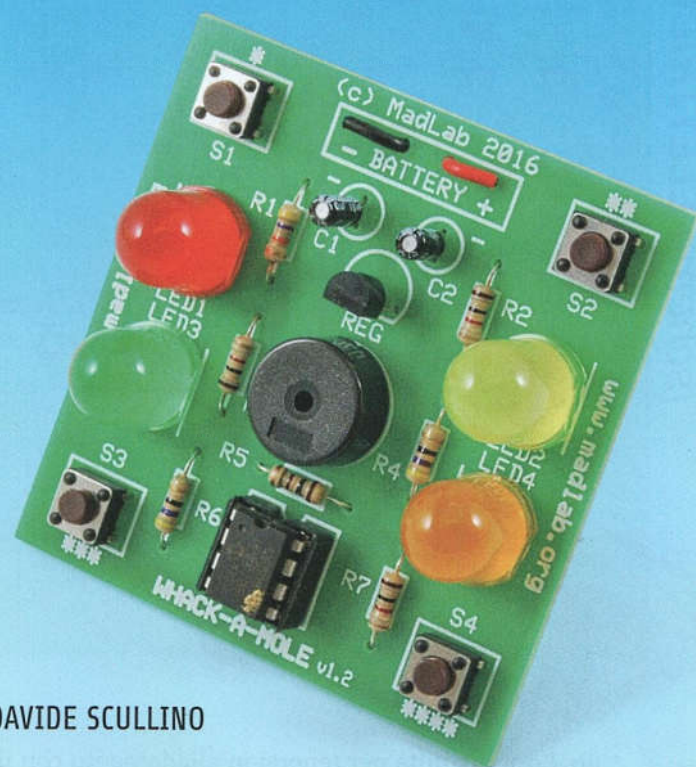
€ 228,00

Cod. **AL2X3005**

Prezzi IVA inclusa.

ACCHIAPPA LA TALPA

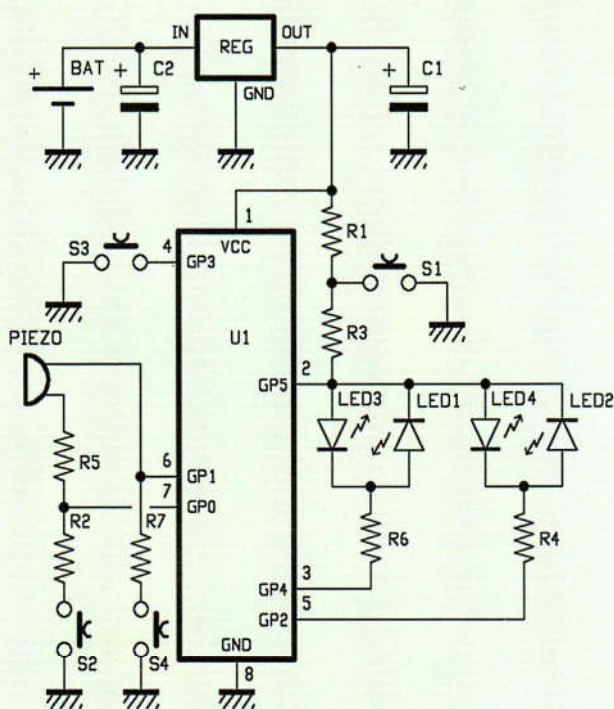
È vero che oggi esistono in commercio giochi altamente tecnologici e che oramai chi vuole giochi elettronici si rivolge alle console di grido o al computer ed allo smartphone, divenuto ormai una piattaforma a 360° con la quale ognuno può portare con sé il proprio mondo, tuttavia realizzare qualche piccolo passatempo tascabile, da tirare fuori quando si attende in una sala d'aspetto o si viaggia (non al volante....naturalmente!!! Ce n'è già troppa di gente che guida messaggiando con lo smartphone...) non è anacronistico. Per questo vogliamo proporvi un piccolo circuito che ci dà l'occasione per fare un po' di didattica e per offrire a chi è stanco di vedere solo Ardui-



di DAVIDE SCULLINO

Simpatico passatempo di abilità, basato su LED e pulsanti che mette alla prova la vostra rapidità di reazione.





no, l'opportunità per tenersi in allenamento con il saldatore.

Il circuito realizza un gioco elettronico di abilità, che serve a testare i riflessi dei giocatori cui è rivolto; a questo gioco cui abbiamo dato il nome di "acchiappa la talpa" pensando che l'accensione di un LED sia come vedere una talpa spuntare dalla propria tana nel terreno e afferrarla (ma a fin di bene...) prima che si rintani.

Insomma, quando si accende un LED la talpa sta spuntando dal terreno e dovete premere il tasto a lei più vicino...

Il nostro passatempo elettronico prevede in realtà di implementare due giochi (dei quali "acchiappa la talpa" è il primo), per ciascuno dei quali sono disponibili due modalità.

SCHEMA ELETTRICO

Analizziamo dunque il circuito che realizza il gioco, che, come ci mostra lo schema elettrico in questa pagina, è semplicissimo perché la logica di gestione è affidata a un microcontrollore: si tratta di un piccolo micro a 8 bit incapsulato in un compatto case DIP a 4 piedini per lato prodotto dalla Microchip e siglato PIC12F508.

Come vedete, il microcontrollore è il cuore del circuito e per far funzionare il tutto gli bastano pochissimi componenti esterni come quattro pulsanti normalmente aperti, altrettanti diodi luminosi, qualche resistenza e un cicalino piezo con elettronica incorporata.

Il tutto si alimenta con una pila a secco da 9 volt grazie al regolatore integrato a tre terminali siglato REG, che è il classico 7805 in versione a bassa potenza (78L05) dal quale otteniamo i 5 volt ben stabilizzati che servono al microcontrollore U1 per funzionare.

Notate che per farci bastare i pochi pin di I/O disponibili, nello schema abbiamo adottato alcuni stratagemmi come il pilotaggio dei LED a coppie in antiparallelo, in modo da poterne accendere uno mantenendo spento l'altro della coppia e viceversa, in base al livello logico; per l'esattezza, tutti i LED impiegati hanno in comune la linea di I/O GP5, però LED1 e LED3 vengono comandati dall'I/O GP4 e LED2-LED4 sono controllati attraverso la linea GP2. Osservate anche che mentre GP4 e GP2 sono inizializzate come output, GP5 viene utilizzata, nel firmware, come I/O bidirezionale e durante la lettura del pulsante viene commutata come input, mentre alternativamente diventa un output, quando lo scopo è accendere uno dei LED del gruppo di quattro; l'accensione dei diodi LED3 e LED4 si ottiene portando a livello logico alto la predetta GP5 e variando i livelli di GP4 e GP2.

Nel dettaglio, quando GP4 è a livello basso si illumina LED3 e se a zero logico ci va GP2 si accende LED4. Per accendere LED1 e LED2, invece, GP5 viene mantenuta a zero logico e si attivano (ovvero si mandano a livello alto) GP2 per LED2 e GP4 per LED1.

Tale modalità di comando implica che i LED non possano essere accesi continuamente ma pulsino (a frequenza elevata per evitare che il nostro occhio lo percepisca) e che LED1 si accenda quando LED3 è spento e viceversa (lo stesso si applica alla coppia LED2-LED4).

Per leggere la condizione del pulsante S1, GP5 viene periodicamente attivata come input privo di pull-up interno; la lettura avviene mantenendo normalmente a livello alto il piedino 2 del microcontrollore tramite la resistenza R1 (che provvede

CARATTERISTICHE TECNICHE

- Tensione di alimentazione: 9 Vcc
- Assorbimento: 60 mA
- 2 giochi con 2 livelli di difficoltà
- Segnalazioni a LED giganti e cicalino
- Interfaccia a pulsanti

quindi al pull-up) e la R3 e verificando quando la pressione del pulsante chiude a masse il nodo tra le due resistenze e quindi fa rilevare il livello logico basso a GP5.

Notate che R3 è stata inserita per evitare che la stessa linea GP5 del microcontrollore venga danneggiata se il pulsante S1 viene mantenuto premuto quando la stessa commuta da input a uscita, il che non è affatto improbabile in quanto l'avvicinamento del piedino 2 dalla condizione di uscita a quella di ingresso avviene molto rapidamente, parecchie volte al secondo, così da rendere impercettibile l'inevitabile pulsazione dei LED, dovuta al fatto che non possono essere accesi costantemente.

Visto che siamo in tema di pulsanti, facciamo notare che l'unico dei quattro ad avere una linea di I/O dedicata è S3, collegato a GP3, che il firmware inizializza come input dotato di pull-up interno. S2 ed S4 sono invece collegati alle linee GP0 e GP1, condivise con il cicalino PIEZO che viene utilizzato dal microcontrollore per dare le segnalazioni previste dalle varie fasi e modalità di gioco.

MODALITÀ DI GIOCO

Il gioco "acchiappa la talpa" prevede due modalità, ciascuna delle quali contempla altrettanti livelli di difficoltà. La selezione si effettua con i pulsanti, dopo aver dato l'alimentazione al circuito; notate che nel momento in cui riceve alimentazione, il circuito comanda il cicalino affinché emetta due brevi note acustiche in sequenza, quindi attendete

queste e poi:

- premete S1 se desiderate la modalità standard; in questa si accenderà un LED a caso e dovete premere rapidamente il pulsante più vicino al LED illuminato; se premete il tasto sbagliato o impiegate troppo tempo, allora avete perso (ciò sarà indicato dal lampeggio alternativo dei LED); notate che il gioco diventa più veloce man mano che superate le varie manche, ovvero ogni volta che vincete, ed avete anche meno tempo per reagire.
- premete S2 per la modalità più difficile del gioco, nella quale a volte si accendono due luci nello stesso istante e dovete quindi premere due pulsanti insieme per confermare di averle identificate (anche in questo caso dovete cercare di vincere più manche possibili).

In tutte e due le modalità di gioco descritte, ogni pressione di un pulsante viene accompagnata dall'emissione di una breve nota acustica da parte del cicalino; quando non riuscite a "stare dietro al gioco" e venite eliminati, la condizione viene segnalata dal suono del cicalino e dal lampeggio alternato dei LED dei due lati del circuito stampato.

Il mouse sposa il videogame

Per i patiti dei videogiochi, Logitech ha realizzato G Pro, un mouse professionale specifico per videogame, nato dall'esperienza fatta lavorando per 2 anni con 50 esperti del settore. Il suo sistema Hero 16K traccia da 100 a 16.000 movimenti al secondo, con un IPS superiore a 400; in questo modo il puntatore si muove nell'esatta posizione del polso in tempo reale. Funziona a batteria ed ha un'autonomia di 48 ore con l'illuminazione accesa e 60 senza. La connessione è wireless e consente

l'aggiornamento dei dati in un solo millisecondo, il che elimina fenomeni di latenza.

Grazie al software in dotazione è possibile programmare il funzionamento dei 6 pulsanti e una memoria interna permette di salvare le preferenze del giocatore. C'è anche la possibilità di impostare la forza di ogni pulsante.

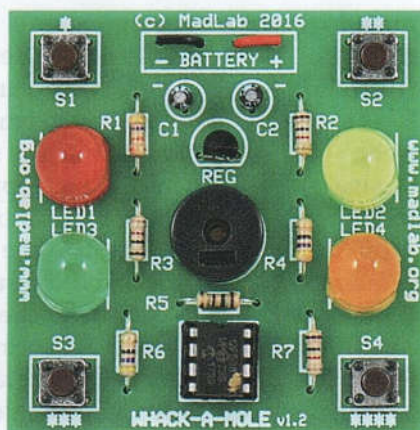
Il tutto pesa 80 grammi ed è incapsulato in un guscio spesso appena 1 millimetro.



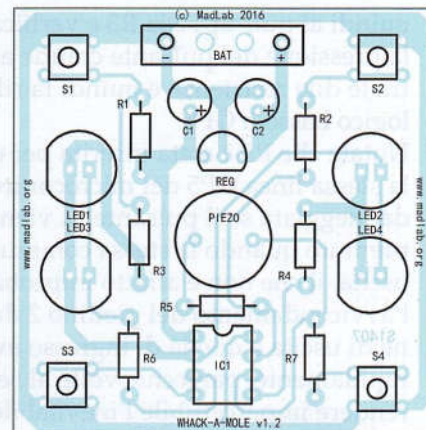
[piano di MONTAGGIO]

Elenco Componenti:

C1: 10 μ F 63 VL elettrolitico
 C2: 10 μ F 63 VL elettrolitico
 R1: 47 kohm
 R2: 1 kohm
 R3: 1 kohm
 R4: 47 ohm
 R5: 100 ohm
 R6: 47 ohm
 R7: 1 kohm
 LED1: LED 10 mm rosso
 LED2: LED 10 mm verde
 LED3: LED 10 mm giallo
 LED4: LED 10 mm arancione
 IC1: PIC12F508-I/P (MF1407)
 S1: Microswitch
 S2: Microswitch



S3: Microswitch
 S4: Microswitch
 PIEZO: Buzzer
 REG: 78L05



Varie:

- Zoccolo 4+4
- Clip batteria 9V
- Circuito stampato S1407 (56x56 mm)

Con il circuito si può anche giocare a un altro gioco chiamato Moley Says, che sostanzialmente è un test di quanto è buona la vostra memoria; anche in questo caso avete la possibilità di scegliere tra una versione standard ed un livello avanzato.

Per giocare a Moley Says dovete:

- premere S3 per avviare la modalità standard, allorché viene visualizzata una sequenza casuale di luci e dovete ripetere la sequenza premendo i pulsanti corrispondenti nello stesso ordine (dopo che la sequenza luminosa è terminata, altrimenti si sbaglia); se avete premuto i tasti nella sequenza corretta, il microcontrollore fa emettere un doppio lampeggio dei LED, mentre se commettete un errore o impiegate troppo tempo, avrete perso (ciò verrebbe indicato dalle luci che lampeggiano alternativamente); anche in questo caso dovete cercare di resistere il più possibile, considerando che le sequenze diventano più complicate man mano che si avanza nelle manche (un lampeggio dei LED in più indica che la sequenza è appena diventata più lunga);
- premere S4 se volete affrontare la versione più difficile; in questo ulteriore livello di difficoltà, a volte vengono accese due luci allo stesso tempo e si devono premere due pulsanti insieme quando si ripete la sequenza.

Per risparmiare l'energia della batteria e aumentare l'autonomia, il microcontrollore prevede il

funzionamento in sleep-mode a bassissimo consumo quando il circuito non viene utilizzato, ovvero quando, finita una manche, per un certo periodo di tempo (circa un minuto); il "risveglio del circuito si ottiene premendo qualsiasi pulsante e viene confermato dall'emissione di due note acustiche e dal lampeggio alternato dei LED dei due lati del circuito. Lo stesso avviene quando si collega la batteria.

REALIZZAZIONE PRATICA

Bene, ciò detto possiamo passare alla costruzione del nostro gioco elettronico, la quale prevede la realizzazione di un circuito stampato per ospitare tutti i componenti occorrenti, da ottenere per fotoincisione una volta scaricate dal nostro sito www.elettronica.in la traccia lato rame e ottenuta da essa la pellicola su carta da lucido o acetato, in scala 1:1. Il circuito stampato è del tipo a singola faccia, cosa che certo apprezzeranno i nostri lettori alle prime armi e quelli non attrezzati per realizzare i complessi PCB a doppia ramatura a fori metallizzati. Il circuito stampato è talmente semplice da poter essere preparato anche per tracciatura diretta anche con l'apposita penna resistente agli acidi e inciso senza ulteriori lavorazioni, ma comunque rendiamo disponibile (e scaricabile) la traccia lato rame, sul nostro sito www.elettronica.in insieme agli altri file del progetto, a chi volesse procedere alla fotoincisione con relativo sviluppo; scaricato i

file, stampatelo su carta da lucido o acetato e otterrete la pellicola per procedere con la fotoincisione. Incisa e forata la basetta, potete montare i componenti, che sono tutti a montaggio tradizionale (THT) a tutto vantaggio di chi non ha la manualità e l'attrezzatura che occorrono alla costruzione di circuiti in SMD; quindi una volta ottenuto il circuito stampato, disponetevi i pochi componenti richiesti a partire da resistenze e condensatori e poi passate ai pulsanti miniatura, allo zoccolo per il microcontrollore, ai condensatori (tutti elettrolitici, per i quali è obbligatorio rispettare il verso indicato) e al regolatore 78L05 (da orientare con la parte piatta rivolta al cicalino e allo zoccolo del micro). Terminate il montaggio con il cicalino piezo (ne serve uno dotato di elettronica interna) e i quattro LED giganti; anche per questi è molto importante la polarità, pertanto assicurarsi che sia corretta, poiché se fosse invertita, durante il funzionamento, qualche LED potrebbe non accendersi in modo corretto.

Per l'alimentazione, saldate nelle piazzole di alimentazione (BAT) i fili rosso (positivo) e nero (al negativo) di una presa volante "a strappo" per pile da 9 volt.

Per l'orientamento dei componenti polarizzati fate riferimento al piano di montaggio che trovate in queste pagine.

Il microcontrollore lo monterete dopo averlo programmato e dopo aver verificato, con l'aiuto del piano di montaggio illustrato in queste pagine, che tutti i componenti siano al loro posto, correttamente orientati e ben saldati, senza baffi di stagno a creare cortocircuiti tra piste attigue.

Una volta completate le saldature, tagliate le eccedenze dei terminali e verificato che tutto sia a posto, inserite il microcontrollore già programmato nel proprio zoccolo, avendo cura di orientarlo come indicato nel piano di montaggio.

Potete quindi utilizzare il vostro circuito, visto che non richiede taratura.

Per programmare il microcontrollore (il firmware potete scaricarlo in formato .asm dal nostro sito www.elettronica.in insieme agli altri file del progetto) vi occorre un programmatore Microchip e l'apposito compilatore: MPLAB IDE va benissimo e il PickKit2 è un programmatore idoneo, ma nulla vi vieta di adoperare quello che desiderate o di autocostruirne uno.

Terminata la programmazione potete collaudare il circuito: fornite alimentazione al circuito tramite la pila da 9 volt.

Ricordate che durante il gioco dovete fare atten-

zione a non toccare la parte posteriore del circuito stampato, cosa che vi verrebbe naturale per poterlo meglio impugnare in modo da premere prontamente i pulsanti; non va fatto perché ciò potrebbe compromettere il funzionamento, in quanto la nostra pelle è un conduttore di elettricità, seppure presenti un'alta resistenza elettrica, con le dita potreste creare percorsi di corrente e collegamenti che andrebbero ad alterare il funzionamento del microcontrollore.

Quindi potete scegliere se isolare il retro del PCB con un foglio di plastica o cartoncino fissato con nastro adesivo ai bordi, ovvero, molto più elegantemente, portare i quattro pulsanti fuori dal circuito stampato, inserire quest'ultimo in un contenitore in plastica dotato di scomparto per la pila o che abbia posto per ospitarla, quindi fissare i pulsanti al pannello frontale e collegarli alle rispettive piazzole del circuito stampato mediante sottili spezzoni di filo elettrico con guaina.

Ricordiamo che malgrado le strategie di risparmio energetico implementate nel firmware, per lunghi periodi di inattività è comunque consigliabile rimuovere la batteria, perché comunque un minimo di assorbimento da parte del circuito c'è sempre e a lungo andare scarica la batteria. In alternativa potete prevedere un interruttore in serie al positivo di alimentazione del filo del portapila.

IL FIRMWARE

Diversamente dalla gran parte dei nostri progetti, qui il microcontrollore è programmato in assembler e il relativo firmware può essere scaricato dal nostro sito web www.elettronica.in insieme agli altri file del progetto. Va compilato con MPASM o IDE Microchip MPLAB IDE in modo da ottenere il file da caricare nel PIC.



per il MATERIALE

Il kit didattico "Acchiappa la talpa" (cod. MLP111IT) è in vendita da Futura Elettronica al prezzo di Euro 9,90 dove è possibile acquistare anche la batteria alcalina da 9V in blister (cod. 6LR61-9V-1) a Euro 1,80. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287 - www.futurashop.it

Meteo sempre sotto controllo!

Sveglia con proiezione dell'ora e del meteo

Tutto sotto controllo grazie alla nuova sveglia con stazione meteo integrata che proietta l'ora e le previsioni meteorologiche sul soffitto o sulla parete. Il display retroilluminato della sveglia visualizza inoltre l'ora (nel formato a 12 o 24 ore), il mese, la data, il giorno della settimana, le previsioni meteo, il grafico della temperatura nell'arco delle 12 ore, la temperatura e l'umidità interna.



€ 36,⁰⁰
Cod. **TP0033**

- ✓ Orologio al quarzo.
- ✓ Calendario perpetuo.
- ✓ Visualizzazione dell'ora 12 e 24 ore.
- ✓ Allarme con snooze.
- ✓ Temperatura interna in °C / °F.
- ✓ Registrazione temperatura e umidità minima e massima.
- ✓ Andamento della temperatura.

- ✓ Proiezione dell'ora e delle previsioni meteo con 4 icone (solleggiato, poco nuvoloso, nuvoloso, piovoso).
- ✓ Retroilluminazione a LED.
- ✓ Alimentazione: 2 batterie da 1,5 Volt tipo AAA (non comprese) oppure tramite adattatore di rete (incluso).
- ✓ Dimensioni: 142x116x65 mm.



€ 49,⁰⁰
Cod. **WH1170**

Stazione meteo con orologio DCF

Stazione meteorologica composta da un'unità base e da un sensore esterno wireless per la rilevazione della temperatura e dell'umidità. Ora e data radiocontrollati, registrazione della temperatura e dell'umidità minima e massima, previsioni e tendenza meteo, modalità di visualizzazione dell'ora 12h/24h, sveglia con snooze, calendario perpetuo e indicatore di batteria scarica.

PORTATA
SENSORE
**50
METRI**



Prezzi IVA inclusa.

Disponibili anche:

Sveglia digitale con campanelli meccanici e doppio allarme



Consente di impostare differenti orari di allarme. Display retroilluminato, funzione snooze.

€ 11,⁸⁰
Cod. **WT705G**

Sveglia radiocontrollata tramite segnale DCF



Display retroilluminato. Visualizza la temperatura interna, data e giorno della settimana. Formato ora: 12/24.

€ 13,⁰⁰
Cod. **TIP020**

Termometro e igrometro con data e ora



Con ampio display LCD visualizza la temperatura interna da -10 °C a +50 °C, l'umidità 10% ~ 99% RH, la data e l'ora.

€ 5,⁰⁰
Cod. **TERMOCLOCK**

Facciamo una panoramica sulle soluzioni adottate nell'interazione tra robot e ambiente e integrate nel concetto di Machine Learning e in quello più ampio di AI.

SENSORI E INTELLIGENZA ARTIFICIALE

..... dell'Ing. DANIELE DENARO

I robot amatoriali o didattici ed in particolare quelli semoventi (detti "rover") utilizzano, per muoversi autonomamente nell'ambiente, sensori che comunemente sono del tipo con uscita digitale; con questo termine intendiamo che forniscono un'informazione sulla semplice presenza o assenza di ostacoli, quindi non sulla distanza da essi. Con questo tipo di interazione con l'ambiente, l'algoritmo di guida difficilmente può avvicinarsi ad un approccio AI (Artificial Intelligence, ovvero Intelligenza Artificiale), rimanendo piuttosto nell'ambito della programmazione tradizionale. Per chiarire questa premessa occorre tentare di schematizzare la natura della AI, per cui introdurremo brevemente questo concetto di cui ultimamente si parla tanto.

INTELLIGENZA ARTIFICIALE: COS'È?

Tutti sanno che un computer può essere idealizzato come uno scatolotto che riceve dei dati, da

tastiera o tramite la selezione con il mouse, e produce il risultato dell'elaborazione di questi dati su schermo o stampante. In questo tipo di utilizzo l'essere umano, che inserisce i dati e riceve il risultato, è la fondamentale interfaccia del computer. Ma in diverse situazioni è possibile, ed anche più utile, eliminare questa mediazione umana e far agire il computer sulla base di input raccolti direttamente dall'ambiente. In questo caso, in genere, il computer produrrà un output che agirà sull'ambiente e l'uomo sarà solo uno spettatore che ne raccoglie i risultati in termini di modifiche fatte alla sua realtà. In genere è proprio questo il futuro immaginato, dall'informatica e dall'elettronica, per il computer. Non è un caso se ultimamente l'elettronica si è molto concentrata sulla produzione di sensori ed attuatori. Un sistema autonomo di questo genere dovrà, per definizione, essere "intelligente", nel senso che dovrà comportarsi in maniera tale da raggiungere uno scopo per

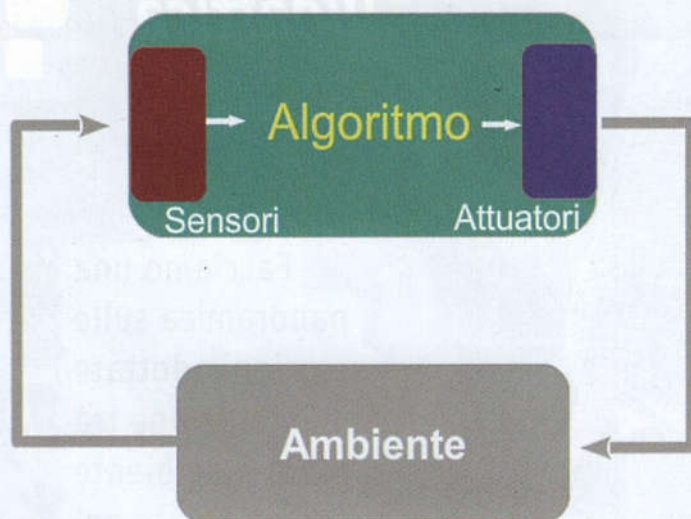


Fig. 1 - Sistema intelligente autonomo.

il quale è stato creato (Fig. 1).

Concentriamoci ora sull'algoritmo che lega funzionalmente l'output all'input di un sistema intelligente autonomo. La capacità di soddisfare più o meno bene gli obiettivi, a prescindere dalla variabilità dell'input, definisce il grado di "intelligenza" dell'algoritmo. In realtà anche un algoritmo molto sofisticato non può ovviare ad una povera qualità dell'input e, in seconda battuta, ad una scarsa qualità dell'output.

Questo vale anche per l'uomo, dove la deprivazione sensoriale non permette uno sviluppo adeguato o confonde il comportamento di un adulto.

Viceversa, un sistema sensoriale sofisticato necessita di un algoritmo sofisticato per esprimere bene le potenzialità del sistema autonomo.

Ecco che un input digitale ON/OFF non permette un comportamento particolarmente intelligente ed occorre passare almeno a sensori con risposta continua (para-analogica). Nel caso di sistemi autonomi se-moventi, per esempio, un sensore di valori di distanza su diversi angoli è il minimo necessario per semplici compiti di orientamento e ricerca; ma avendo un tal genere di sensori, sorge subito il problema di interfacciarli con un algoritmo che gira su un elaboratore digitale.

Dunque, come gestire questa ricca quantità di informazioni? Non sono praticabili serie infinite di "if ... then" e perciò occorre utilizzare altri paradigmi. Ed è qui che si comincia a parlare di Intelligenza Artificiale.

Occorre innanzitutto fare una distinzione di base fra questi due tipi di algoritmi: algoritmi programmati ed algoritmi che imparano.

ALGORITMI PROGRAMMATI

I sistemi a logica fuzzy sono algoritmi che in un certo senso fanno in modo da adattare la sensoristica

continua al mondo digitale del computer. La logica fuzzy se la cava simulando un computer virtuale con logica non booleana ma continua, ovvero facendogli trattare variabili che possono avere un grado di verità continuo tra 0 ed 1 (vedi rivista 169). È stata utilizzata con successo anche nei microcontrollori, perché permette di gestire sensori continui in modo efficiente anche con hardware limitato.

Tale tecnica è il modo più semplice per realizzare, ad esempio, sistemi di controllo automatico semplificati o per comandare robot semoventi con sensori complessi e a risposta continua.

Siamo dunque entrati nel campo dell'AI? Non è detto, perché dipende dalla complessità dell'algoritmo che utilizza la *fuzzy logic*. Definire il settore dell'AI è infatti complesso. In prima approssimazione, possiamo dire che un sistema rientra nell'AI se il suo comportamento simula il comportamento umano (vedi "Test di Turing"). Ma a questo punto si tratta di definire il "comportamento umano".

Ce la possiamo cavare affermando che un comportamento umano è sostanzialmente adattativo, cioè capace di adattare la propria risposta al variare della risposta ambientale.

Esempi di applicazioni di AI, sono i sistemi esperti e i programmi di esplorazione euristica; si tratta di algoritmi che trattano comunque dati digitali, quindi poco adatti a gestire sensori che forniscono segnali analogici. I sistemi esperti tentano di simulare il comportamento di un essere umano descrivendo una grande serie di regole (booleane del tipo "if ... then" dette "basi della conoscenza"), ricavate interpellando gli esseri umani per ogni possibile situazione di input. Chiaramente questo approccio è limitato ad argomenti molto circoscritti della realtà ed è stato uno dei primi tentativi di AI che ha ottenuto qualche significativo risultato ma che ormai è in fase di superamento. Gli algoritmi euristici hanno ambiti ristretti e specialistici come per esempio i programmi di gioco.

ALGORITMI CHE IMPARANO

Gli algoritmi programmati prevedono l'esistenza di un essere umano che costruisca ed implementi la relazione tra input ed output, spesso molto complessa. Al contrario, gli algoritmi che imparano prevedono che l'essere umano costruisca ed implementi un sistema che sia capace di rappresentare autonomamente questa relazione, in base ad alcune direttive di massima la cui natura differenzia i vari tipi di apprendimento. Poiché l'apprendimento è caratteristico dell'essere umano (ma anche degli altri essere viventi) possiamo dire con maggiore sicurezza che

entriamo nel campo dell'AI. Anche questa volta dobbiamo fare una suddivisione fra due tipologie: *apprendimento supervisionato* e *apprendimento non supervisionato*.

APPRENDIMENTO SUPERVISIONATO

Questi algoritmi, per imparare, hanno bisogno di una serie consistente di esempi input/output; in base a questi esempi costruiscono una corrispondenza anche complessa ed in genere statistica tra i valori dei sensori e il corretto output. La Fig. 2 propone un esempio di addestramento supervisionato. Quindi, dopo l'addestramento, possono rispondere in maniera appropriata a qualunque insieme di valori di input oltre a quelli degli esempi. Questi algoritmi si rifanno, in genere, a sistemi a "reti neurali". Anche i sistemi "Deep Learning", molto di moda attualmente, sono di questo tipo. Il riconoscimento di pattern e di immagini è l'applicazione più comune. Rispetto alle prime reti neurali, i sistemi moderni hanno inserito una serie di elaborazioni e filtri che permettono alle reti neurali di estrarre meglio la *conoscenza* dagli esempi. È evidente che in queste applicazioni, fermo restando che l'input sensoriale rimane complesso, l'output può anche essere un avviso all'interfaccia umana e non un azionamento. Il punto di forza di questo tipo di algoritmi basati sulle reti neurali è l'efficiente modalità che aggiusta i parametri al verificarsi di un errore sull'output; tale modalità è chiamata "Error Back Propagation".

APPRENDIMENTO NON SUPERVISIONATO

Adesso le cose si fanno più difficili: non c'è più un essere umano che addestra e quindi il sistema deve acquisire tutte le informazioni dall'ambiente, compreso l'errore che commette ad ogni passo. Dato

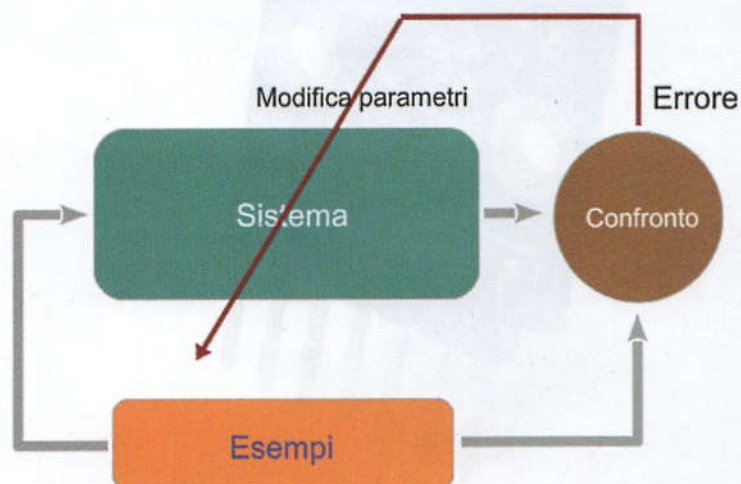


Fig. 2 - Addestramento supervisionato.

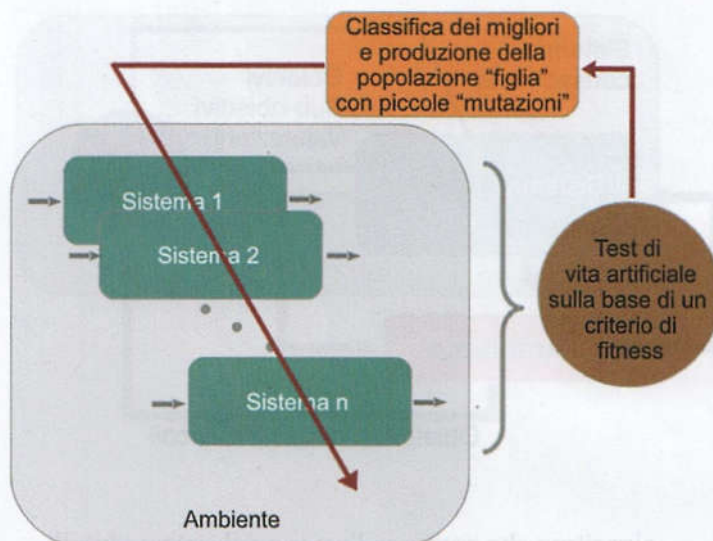
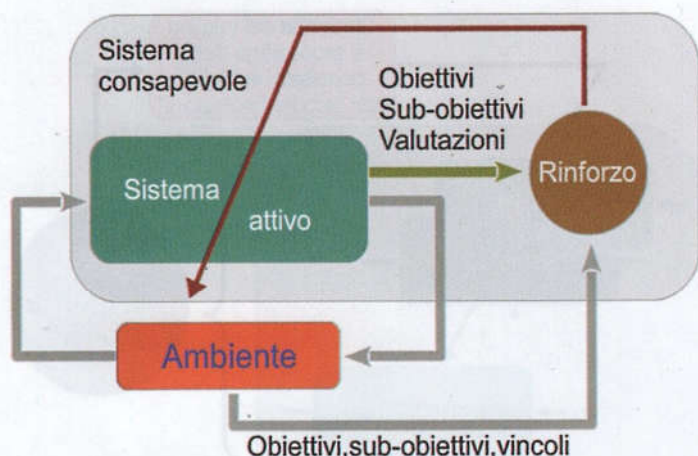


Fig. 3 - Apprendimento con algoritmo evolutivo.

che gli serve per correggere la propria risposta allo scopo di migliorare le proprie performance. Ma sulla base di quale criterio si definisce l'errore o la bontà della risposta, se non sono più disponibili gli esempi? Per chiarire questo aspetto facciamo riferimento alla natura: come fa un essere vivente a modificare il suo comportamento per adeguarlo alle condizioni ambientali con la massima efficienza? In due modi: con l'evoluzione e con il rinforzo. La prima agisce in tempi lunghissimi e a livello di popolazione, mentre il secondo, durante la vita e sul singolo individuo, cercando di soddisfare uno o più stimoli interni, "cablati" dall'evoluzione. In genere fame e spinta alla procreazione.

L'imitazione dell'evoluzione naturale ha prodotto gli algoritmi genetici o evolutivi (Fig. 4) che prevedono il test di una popolazione di sistemi intelligenti con la selezione dei migliori in base ad un criterio di "fitness" stabilito a priori. La popolazione di partenza è prodotta con parametri casuali per ogni singolo sistema: per esempio, pesi casuali delle sinapsi della rete neurale che implementa l'algoritmo. Dai migliori viene generata una successiva popolazione di figli applicando piccole mutazioni casuali ai parametri dell'algoritmo. Popolazione dopo popolazione, il comportamento degli individui migliora anche se non è garantito il raggiungimento del comportamento ottimale. Però tale tecnica è in genere utilizzata in un mondo virtuale con individui simulati, per ragioni pratiche; ma nulla vieta di riutilizzare gli algoritmi risultati migliori per gestire un sistema fisico, chiaramente a patto che la simulazione di vita artificiale sia stata soddisfacentemente realistica.

Però l'essere vivente ed in particolare l'uomo, impara per tentativi ed errori anche durante la vita; in questo paradigma occorre perciò immaginare un



algoritmo che corregga l'azione del sistema intelligente, in base ad uno o più criteri di soddisfazione. Questi algoritmi sono detti di "rinforzo" perché cercano di valorizzare le azioni che avvicinano all'obiettivo (o agli obiettivi) mentre cercano di far dimenticare le azioni poco produttive. Il tutto per ognuno dei pressoché infiniti possibili valori dell'input (o se vogliamo degli stati del sistema). È evidente che occorrono algoritmi che ancora una volta estraggano e sintetizzino classi di azioni per classi di input sensoriale. Sono i sistemi più complessi ed ancora non risolti in modo generale ed ottimale. Infatti oltre al sistema agente c'è bisogno anche di un sistema che valuti la situazione in termini di raggiungimento di obiettivi. Lo sviluppo di questi algoritmi potrebbe portare, probabilmente un domani, addirittura all'auto coscienza dei sistemi intelligenti autonomi, ovvero alla cosiddetta "AI forte", che si distingue dalla "AI debole" che è relativa al comportamento intelligente per compiti limitati, come quella realizzata dalle reti neurali, Deep Learning, ecc.

SENSORI TOF

Tornando al nostro "rover", senza scomodare la complessa gestione delle immagini di una eventuale telecamera posizionata su di esso, già un sistema di tipo "radar" basterebbe a fornire una ricca informazione per un movimento intelligente. Un metro ad ultrasuoni di costo accettabile non ha risoluzione e orizzonte necessario, perché bisognerebbe montarlo su un sistema ruotante, tipo servo, per realizzare una scansione continua; oppure metterne molti a coprire un ampio angolo.

Ultimamente, però, sono apparsi nuovi tipi di sensore di distanza basati sul "tempo di volo" (ToF: Time of Flight) di un raggio laser, che hanno un prezzo relativamente basso (< 20€) e sono più precisi di un sistema ad ultrasuoni, anche perché l'angolo di misura può essere ben più ristretto. Un primo prodotto è quello basato sull'integrato VL6180

Fig. 4 - Apprendimento per rinforzo.

(della ST) di cui Futura Elettronica distribuisce il circuito di test (breakout board, mostrata nella Fig. 5) con il codice "BREAKOUT017".

La piccola scheda non fa altro che adattare la tensione di alimentazione e i livelli logici del bus I²C; può essere alimentata anche a 5V e i livelli logici del bus di comunicazione possono essere anch'essi a 5V, mentre l'integrato, di suo è compatibile con i livelli 0/3V. Il collegamento con Arduino o altro microcontrollore avviene tramite il bus seriale I²C e sfruttando una libreria predisposta scaricabile dal sito Internet www.futurashop.it.

La libreria rende il funzionamento molto semplice e, una volta inizializzato il sensore, basta chiedergli la distanza che rileva in quel momento. La distanza massima può arrivare a quasi 15cm ed è garantita una certa immunità alle variazioni di luminosità ambientale ed alle variazioni di riflettanza degli oggetti. L'emissione del laser è sull'infrarosso (850nm) ed il suo angolo di rilevamento è di circa 25°, mentre il tempo di risposta è al massimo di 15ms.

Per provare ad avere una rilevazione continua su 180°, possiamo posizionare il sensore su un servo e fare una scansione bidirezionale. Purtroppo un servo troppo economico (GS-9018) non ha un corretto posizionamento assoluto e la scansione cambia asseccando del verso di rotazione. Si può allora provare ad utilizzare un motore passo-passo, l'economico

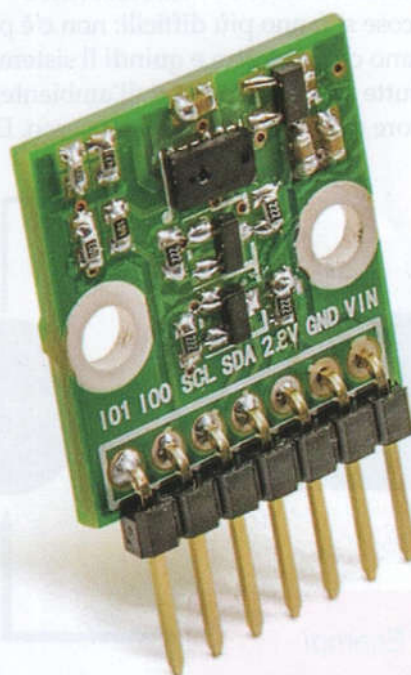


Fig. 5 - ToF su breakout board.

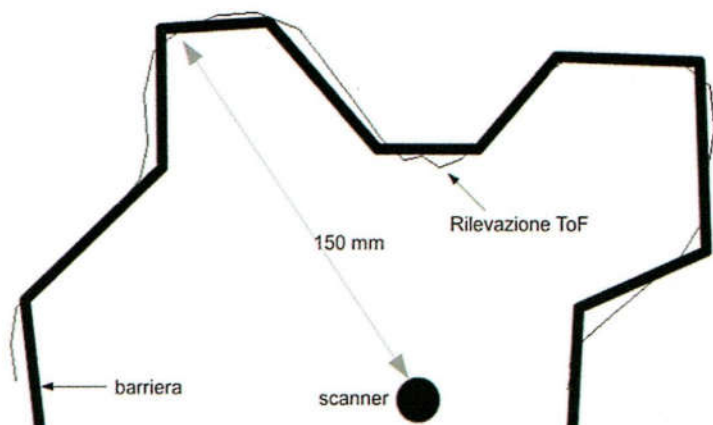


Fig. 6 - Rilevazione ToF su servo o stepper (mediata e corretta).

SM28BYJ (codice SM2003DB), che è anche più silenzioso ed affidabile per una scansione continua. Il risultato è stato migliore, ma rimane il problema del posizionamento assoluto di partenza: infatti, non avendo un "feedback" di posizione come i servo, lo spostamento va gestito da software a partire dalla posizione di start che deve essere inizializzata a mano. La verifica dello "scanner" è stata fatta utilizzando una sagoma di cartone rappresentata dalla linea spessa in Fig. 6. Nella Fig. 7 potete invece vedere la differenza tra l'utilizzo di un servo GS-9018 e di uno stepper come SM28BYJ per la scansione. Comunque anche nel primo caso applicando un

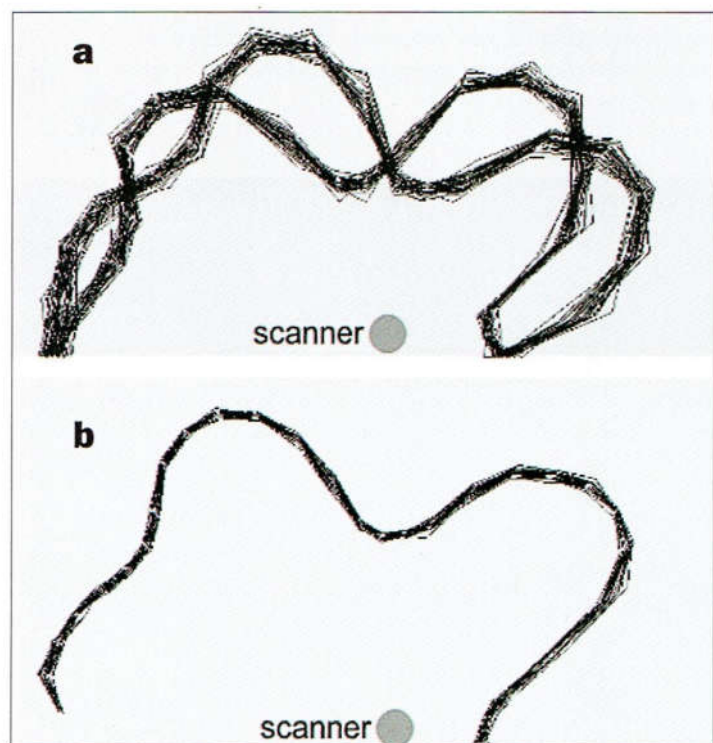


Fig. 7 - a) scansione con servo economico
b) scansione con stepper-motor.

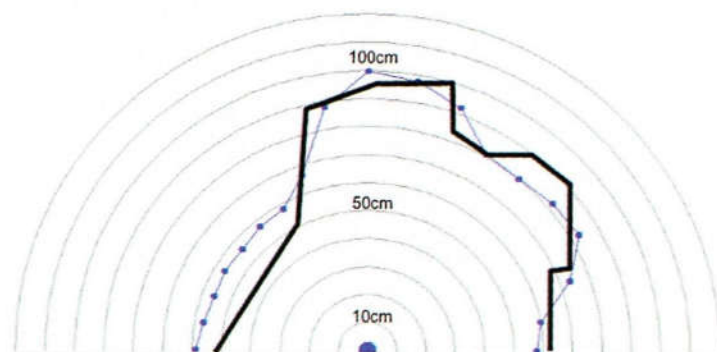


Fig. 8 - Scansione con ToF VL5310X.

"offset" sulla scansione di ritorno si ottiene un buon risultato. In ambedue i casi è opportuno scansionare con angoli $\leq 15^\circ$ ed applicare una media.

Questo scanner così composto potrebbe essere utilizzato su un "rover" a due ruote molto piccolo, dato il range limitato che lo contraddistingue. In sostanza, per esempio un robottino didattico da scrivania, a patto di mantenere una piccola dimensione anche per lo scanner.

Il secondo prodotto ToF è più sofisticato e raggiunge in teoria i 150/200cm. Con una tale visibilità degli ostacoli è pensabile un utilizzo su un semovente da pavimento o da terreno. L'integrato che lo implementa ha la sigla VL5310X (sempre della ST) e la breakout board corrispondente (sempre prodotta dalla Futura Elettronica) è praticamente identica a quella descritta poco fa, ma è siglata "BREA-KOUT018". Questa scheda è già stata utilizzata nel progetto del rilevatore di ausilio al parcheggio proposto nella rivista n° 221; in quell'occasione è stato utilizzato come unico punto centrale di rilevazione. Vediamo ora come potrebbe funzionare nell'ambito di una scansione continua. Il laser emesso è su un infrarosso un po' più spinto (940nm), ma soprattutto ha un microprocessore integrato più potente per poter elaborare più rilevazioni, fornendo un risultato ripulito dai disturbi. Infatti i tempi di risposta sono più alti, circa 30ms di base, fino ad arrivare anche a 300ms nella modalità "High accuracy". La libreria è diversa dalla precedente e permette maggiori possibilità di messa a punto. Infine l'angolo di rilevazione è anche questa volta di 25° .

Le prove effettuate utilizzando la libreria hanno però dimostrato alcuni limiti. Innanzi tutto un test come metro ha rilevato una soddisfacente misura solo fino alla distanza di poco più di un metro. Con un errore di $\pm 3\text{cm}$ in questo range. Anche la configurazione "Long Range" non ha cambiato la situazione. In ogni caso applicando la scansione a 180°



Fig. 9 - Lidar YDLIDAR X4.

limitata ad una distanza di 120 cm si è evidenziata una maggiore approssimazione dei risultati rispetto al precedente ToF. In Fig. 8 è mostrato un esempio di scansione per il rilevamento di un perimetro rappresentato dalla linea più spessa.

Con questi presupposti, una navigazione può contare su una visione della realtà frontale solo approssimativa e poco precisa; siccome l'integrato ha un largo margine di configurazione, non è escluso che un approfondito studio e regolazione della libreria e dei parametri possa migliorare la resa.

Ma un prodotto più completo e utilizzabile è quello descritto nel prossimo paragrafo.

SENSORI LIDAR

Lo sviluppo della ricerca finalizzata a creare auto a guida autonoma ha portato sul mercato una serie di prodotti con funzionalità radar. Tra questi i LIDAR (*Laser Imaging Detection and Ranging*) sono diventati molto popolari ed hanno cominciato ad essere por-

tati anche in campo hobbistico. Sono sostanzialmente dei ToF più sofisticati ed integrati con un sistema di scansione.

Uno di questi è il YDLIDAR X4 della "EAI" (<http://www.eaibot.com/>) visibile nella Fig. 9 e distribuito da Futura Elettronica con il codice FR704. Ha destinazione amatoriale ed ha un costo orientativo poco maggiore di 100€. Ha dimensioni di circa 7X10 cm con una altezza di circa 5 cm. Un motore passo-passo silenzioso fa ruotare la testa che supporta il laser emettitore ed il ricevitore. Ha una risoluzione angolare di mezzo grado ed un range da 10 cm a 10 m, con una precisione maggiore di 1 cm (mediamente 5 mm). Ruota con una velocità di circa 1,5 giri/sec, quindi compie circa 5.000 scansioni al secondo. Comunica tramite seriale a 128.000 baud tramite un protocollo applicativo proprietario. Viene fornita comunque una libreria per il suo utilizzo.

Come si può vedere dalla Fig. 10, la scansione anche a distanze metriche fornisce una pianta plausibile dell'ambiente scansionato.

In ogni caso occorre avere a disposizione algoritmi e software che possano utilizzare efficacemente questa quantità di input sensoriali di tipo analogico. In un articolo di prossima pubblicazione cominceremo ad affrontare la problematica partendo da sistemi leggeri da utilizzare su hardware limitato come Arduino, fino ad impegnare le maggiori capacità di calcolo presenti in Raspberry Pi. In particolare, nel prossimo articolo, parleremo di Reti Neurali e di una semplice realizzazione per guidare un rover come Ardusumo.

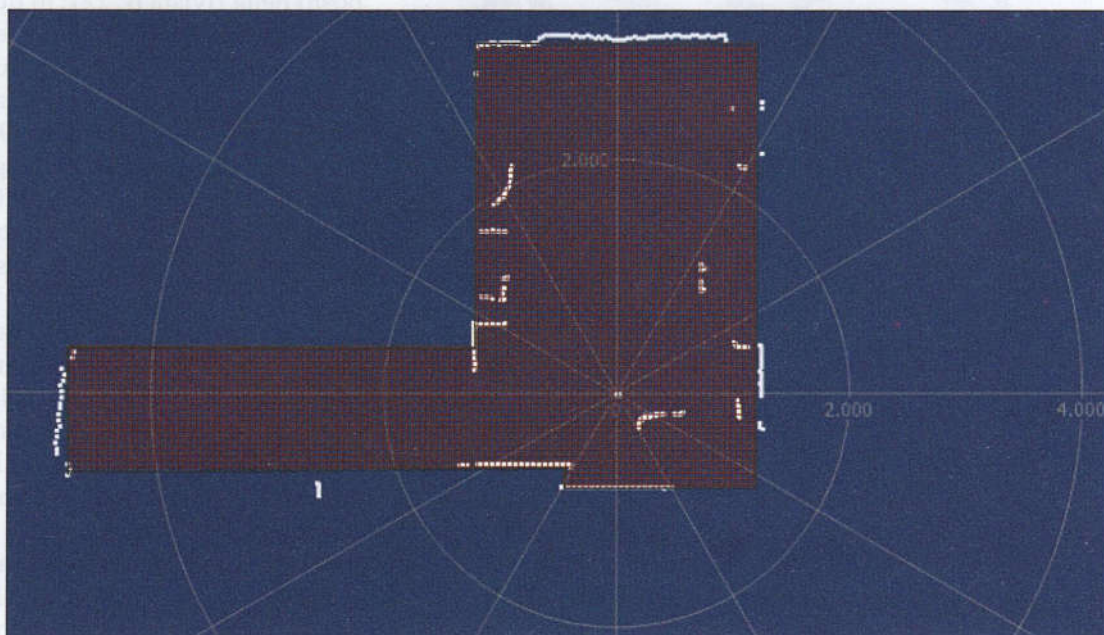
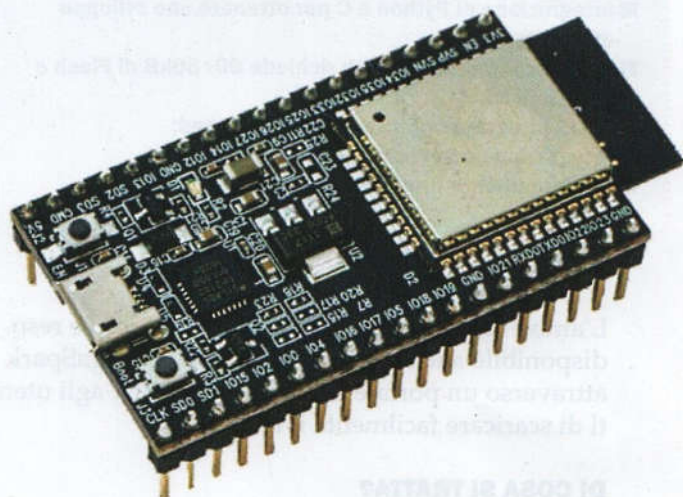


Fig. 10 - Esempio di scansione con YDLIDAR X4 di una stanza+corridoio +mobili.

ZERYNTH®
Your Ideas. Embedded

How to get started
with the ESP32 and
Zerynth Studio



ZERYNTH STUDIO: PROGRAMMA I MICRO IN PYTHON

Ecco Zerynth Studio, la suite per applicazioni IoT
che permette la programmazione in Python dei microcontrollori a 32 bit.

..... a cura della REDAZIONE

Zerynth è un'innovativa startup italo-americana che sviluppa ottimo software middleware per la programmazione di applicazioni dell'Internet delle Cose (IoT); con il termine "middleware" si intende uno o più programmi che fanno da intermediari tra l'hardware e un'applicazione software: nel nostro caso si tratta di mediare tra il linguaggio macchina di varie famiglie di microcontrollori e un linguaggio di programmazione evoluto, che di suo non potrebbe essere interpretato, quale è Python. A stretto rigore anche il C o il PicBasic debbono

essere interpretati per caricare i relativi firmware nella memoria di programma dei microcontrollori, quindi un software middleware c'è sempre, tuttavia la novità proposta da Zerynth è che si tratta di un interprete di python, che consideriamo molto "azzeccato" in quanto oggi tale linguaggio è tra i più utilizzati e oltretutto dispone di strumenti e funzionalità avanzate che nel C non sono disponibili. Quindi è una novità assoluta ed anche una vera rivoluzione nel mondo della programmazione dei microcontrollori.

CARATTERISTICHE DELL'AMBIENTE

Zerynth integra un sistema operativo in tempo reale (RTOS) che consente anche agli sviluppatori di utilizzare tutte le funzionalità dell'hardware incorporato, se necessario.

- Integrazione di Python e C per ottenere uno sviluppo efficiente;
- Ingombro ridotto: Zerynth richiede 60÷80kB di Flash e 3÷5kB di RAM;
- RTOS integrato con supporto multithread;
- Integrazione con i principali servizi cloud;
- Aggiornamento firmware Over-The-Air.

L'ambiente di sviluppo IoT Zerynth Studio è reso disponibile anche alla community di DesignSpark attraverso un portale unico, che permette agli utenti di scaricare facilmente il toolkit.

DI COSA SI TRATTA?

Zerynth fornisce una suite di strumenti software che facilitano la programmazione di microcontrollori a 32 bit e la loro connessione alle principali infrastrutture cloud, consentendo una rapida implementazione dell'IoT. La suite comprende Zerynth Studio, un ambiente di sviluppo integrato multiplatforma che funziona su Windows, Linux e Mac OS X.

La suite di strumenti include anche un compilatore, un debugger e un editor di codice avanzato; inoltre fornisce tutorial ed esempi di progetto e consente a progettisti e sviluppatori di utilizzare il linguaggio di programmazione Python o una versione ibrida C/Python.

Con Zerynth Studio i progettisti possono iniziare a programmare dopo aver installato le VM (Virtual Machines) di Zerynth su uno dei numerosi microcontrollori supportati disponibili dai principali produttori, tra cui Espressif, Nordic Semiconductor, NXP, Microchip/Atmel e STMicroelectronics, collegandoli ai principali servizi cloud di Amazon, Google, IBM o Microsoft.

La suite DesignSpark Zerynth Studio è scaricabile da luglio 2018 dal sito della community DesignSpark, che offre già una serie di strumenti software gratuiti, risorse e informazioni per aiutare i progettisti a sviluppare i loro progetti di elettronica, portandoli rapidamente dall'idea alla creazione di prototipi.

QUALCOSA SU ZERYNTH

L'enorme crescita dell'Internet delle cose e la simultanea esplosione dell'interesse di maker e professionisti per le schede di prototipazione, ha

spinto la ricerca a ripensare anche il modo per rendere il mondo embedded facilmente accessibile a un vasto pubblico e non solo agli esperti: designer, artisti, ma anche ingegneri senza un solido background nella programmazione di basso livello. Zerynth è una soluzione per designer e professionisti IoT che devono concentrare i propri sforzi sulla progettazione e sulle idee di progetto, senza preoccuparsi di aspetti come la programmazione a basso livello di particolari schede a microcontrollore, driver di dispositivi e così via.

L'ECOSISTEMA ZERYNTH

Zerynth consente di programmare molti microcontrollori a 32 bit (ad esempio ESP32, ESP8266, ARM Cortex M0+, ARM Cortex M3, ARM Cortex M4 e altri) usando Python o un ibrido C/Python.

Zerynth ha fornito agli utilizzatori dei moduli Node MCU ESP32 (specificamente creati per realizzare applicazioni connesse), uno strumento per semplificare lo sviluppo di applicazioni IoT incorporate in Python ed è anche uno strumento di sviluppo ufficiale di terze parti di Microchip.

Zerynth consente di collegare i microcontrollori ai principali servizi cloud come Amazon Web Services, Google Cloud Platform, IBM Cloud, Microsoft Azure e altri.

Zerynth garantisce un rapido sviluppo e un'efficace integrazione con sensori, attuatori e servizi cloud grazie a un vasto elenco di librerie Python open-source. I tool disponibili sono tanti e li trovate elencati qui di seguito.

- Zerynth Studio, che è un IDE multiplatforma gratuito e una Toolchain per lo sviluppo di applicazioni Python o ibride C/Python e per la gestione di schede; include un compilatore, un debugger e un editor di codice avanzato, oltre a tutorial e progetti di esempio.
- Zerynth Virtual Machine (Zerynth VM) è il cuore della suite Zerynth. Con essa diventa possibile sviluppare script Python 3, che sono praticamente indipendenti dalla piattaforma hardware selezionata, consentendo una elevata riusabilità e integrazione del codice. Zerynth VM è stato sviluppato da zero con l'obiettivo di portare Python nel mondo embedded con il supporto per la compatibilità multi-thread e cross board. Zerynth VM supporta la maggior parte delle funzioni di alto livello di moduli, classi, multi-threading, callback, timer ed eccezioni di Python. Oltre a questi, consente di utilizzare funzionalità hardware personalizzate come interrupt, PWM e I/O digitali.

- Zerynth Advanced Manager (ADM) espone un'interfaccia basata su API per semplificare l'inoltro dei dati cloud, gli aggiornamenti FOTA (Firmware Over-The-Air) e il Remote Procedure Calling.
- Zerynth App è un'applicazione mobile che consente agli utenti di monitorare oggetti basati su Zerynth da dispositivi mobili per Android o iOS The Zerynth App.

Sul piano della licenza di utilizzo, Zerynth Studio è open-source e il suo download è libero.

Invece il tool Zerynth Virtual Machine non è open e viene distribuito come file binario; la relativa licenza viene concessa per ogni specifico microcontrollore con chip ID univoco dopo che l'utente ha creato un account Zerynth.

Alla creazione dell'account l'utente riceve in omaggio un "Welcome pack" di 10 licenze Zerynth VM con FreeRTOS (cinque "Starter" e cinque "Premium") da utilizzare su ciascuna delle schede supportate.

Ulteriori licenze saranno concesse dietro pagamento, secondo la policy di Zerynth.

PROGRAMMARE IL MODULO ESP32 IN PYTHON

Il modulo ESP32 della Espressif è uno dei dispositivi di prototipazione low-power più popolari tra quelli dotati di connettività WiFi e Bluetooth offerti dal mercato.

Ci sembra quindi il supporto hardware ideale con cui iniziare a lavorare con Zerynth Studio ed è per questo che vi proponiamo un esempio applicativo basato su di esso, in grado di farvi toccare

con mano come programmare microcontrollori in Python con Zerynth Studio sia facile, flessibile e potente. Soprattutto con una scheda dedicata basata su ESP32 della Espressif Systems, marchiata Zerynth.

DOWNLOAD DELL'AMBIENTE

Partiamo subito con il progetto compiendo il primo passo, ossia il download di Zerynth Studio, dalla pagina web www.zerynth.com/zerynth-studio/, la quale, come mostra la Fig. 1, prevede icone differenti per le piattaforme previste, ossia Windows, Linux e Mac OS.

Scaricate quella che vi serve facendo clic sull'icona corrispondente e seguite le istruzioni a video (salvate il file nel percorso desiderato, scompattate ed eseguite). A fine installazione siete pronti per lavorare.

COLLEGHIAMO, REGISTRIAMO E VIRTUALIZZIAMO LA SCHEDA

Il NodeMCU ESP-32S è dotato di un chip convertitore da seriale a USB, che consente di programmare la scheda e di aprire l'UART del modulo ESP32.

A seconda del sistema operativo che avete sulla macchina di sviluppo, può essere necessario scaricare dei driver: succede tipicamente con MacOS o Windows (con Linux, il problema non si pone, perché non li richiede); se vi servono i driver, possono essere scaricati dalla pagina di documentazione ufficiale della Espressif (www.espressif.com/en/support/download/other-tools). Nei sistemi Linux, il NodeMCU ESP-32S dovrebbe funzionare immediatamente.

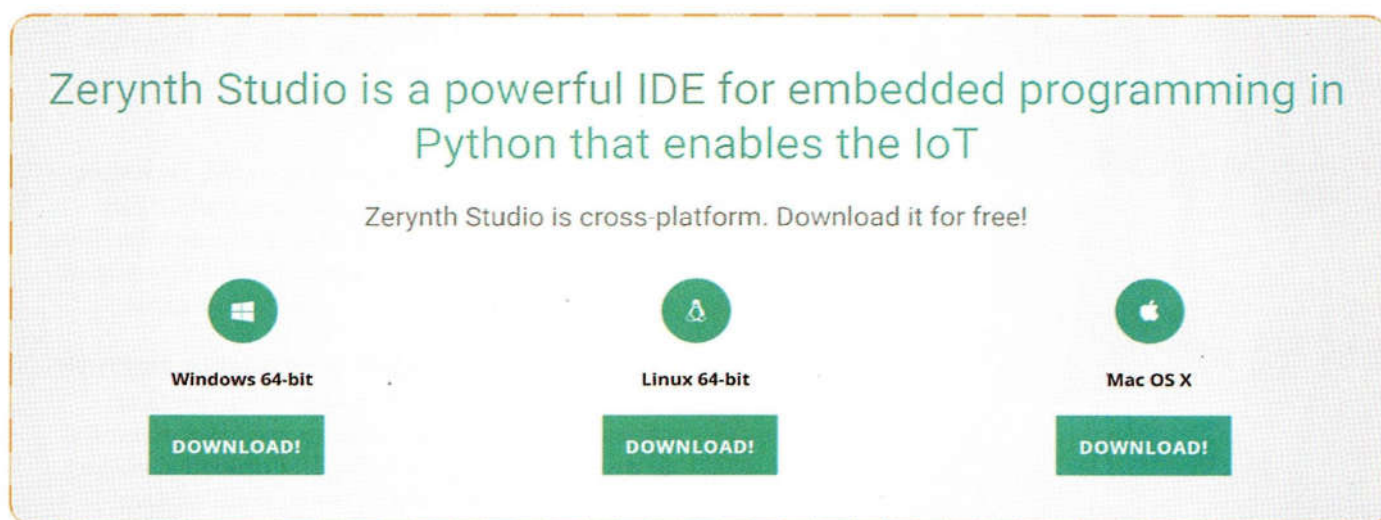


Fig. 1 - La pagina di download di Zerynth Studio.

|| NodeMCU ESP-32S

La scheda di sviluppo e prototipazione ESP-32S che utilizzeremo è sostanzialmente un modulo ESP-WROOM-32, il cui cuore è l'ESP32 della Espressif Systems che implementa i link wireless WiFi e Bluetooth e la connessione Ethernet, supportando la gestione Low Power.

Il tutto è governato da un microcontrollore Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6.

Le caratteristiche di questo modulo sono:

- tensione di lavoro interna: 3,3V;
- tensione d'alimentazione esterna Vin: 7÷12V;
- Digital I/O (DIO): 28;
- Input analogici(ADC): 8;
- Output analogici (DAC): 2;
- UART: 3;
- SPI: 2;
- I²C: 3;
- Flash Memory: 4 MB;
- SRAM: 520 KB;
- Clock Speed: 240 MHz;
- WiFi: IEEE 802.11 b/g/n/e/i.

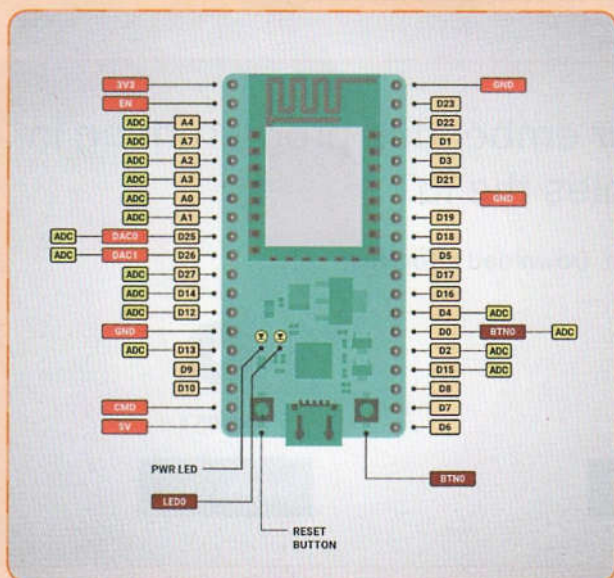
Per quanto riguarda la sezione WiFi, integra lo switch TR, il BalUn (Balancer-Unbalancer), LNA, amplificatore di potenza e matching network. Viene supportata la codifica di sicurezza. WEP o WPA/WPA2, ma è anche prevista la possibilità di connettersi in reti aperte.

Su tutti gli output digitali è possibile inviare il PWM.

Va inoltre tenuto conto che per come funziona la board, D0, D2, D4, D12, D13, D14, D25, D26, D27 possono essere letti solo se il modulo WiFi non è attivato.

La figura in questo riquadro mostra la piedinatura della board, basata su ESP32.

Il Node MCU ESP-32S si può alimentare attraverso il connettore USB Micro B di cui dispone, oppure direttamente dal pin Vin: un'apposita logica seleziona automaticamente la sorgente di alimentazione. Alimentando la scheda da Vin, la tensione fornita (continua) dev'essere di valore compreso fra 7 e 12 volt e dev'essere possibilmente stabilizzata.



Una volta connesso su una porta USB, se i driver sono stati installati correttamente il dispositivo viene riconosciuto dall'ambiente Zerynth Studio.

I prossimi passi che dovete compiere sono:

- selezionare il NodeMCU ESP-32S sulla barra di gestione dei dispositivi (se in essa esistono più versioni, specificare quella che state usando);
- registrare il dispositivo facendo clic sul pulsante "Z" di Zerynth Studio;
- creare una macchina virtuale per il dispositivo facendo clic sul pulsante "Z" per la seconda volta;
- virtualizzare il dispositivo facendo clic sul pulsante "Z" una terza volta.

Se occorre, potete trovare maggiori informazioni su questo processo alla pagina web https://docs.zerynth.com/latest/official/board.zerynth.nodemcu_esp32/docs/index.html del progetto Zerynth.

CREATE IL VOSTRO PRIMO SCRIPT PYTHON

Bene, effettuata la virtualizzazione possiamo passare alla parte più “succulenta” di questo articolo, vale a dire alla creazione del nostro primo script in Python per il NodeMCU ESP-32S, che a questo punto possiamo ritenere pronto per essere programmato, eseguendo l’upload dello script Zerynth desiderato.

Per creare il nostro primo script Python script con Zerynth Studio, semplicemente copieremo il codice di esempio "Hello Zerynth" eseguendo questa procedura:

- aprete il browser Esempi facendo clic sull'icona "lampadina" nel pannello di sinistra di Zerynth Studio;
- dalla struttura ad albero che si apre, selezionate l'esempio "Hello Zerynth" (Fig. 2) e clonatelo cliccando sul pulsante dedicato a tale operazione.

In questa fase Zerynth Studio converte l'esempio in un nuovo progetto, dandoci la possibilità di modificare il titolo, la descrizione e la cartella dove salvare questo nuovo progetto; fatto ciò, possiamo cliccare sul pulsante "Crea" e il gioco è fatto!

VERIFICARE E CARICARE IL PROGETTO ZERYNTH

Passiamo adesso allo step successivo, ossia al caricamento del progetto software nella memoria di programma del Node MCU; l'upload si esegue mediante i pulsanti che si trovano sulla barra situata nella parte superiore sinistra della finestra di lavoro di Zerynth Studio, che nella **Fig. 3** sono

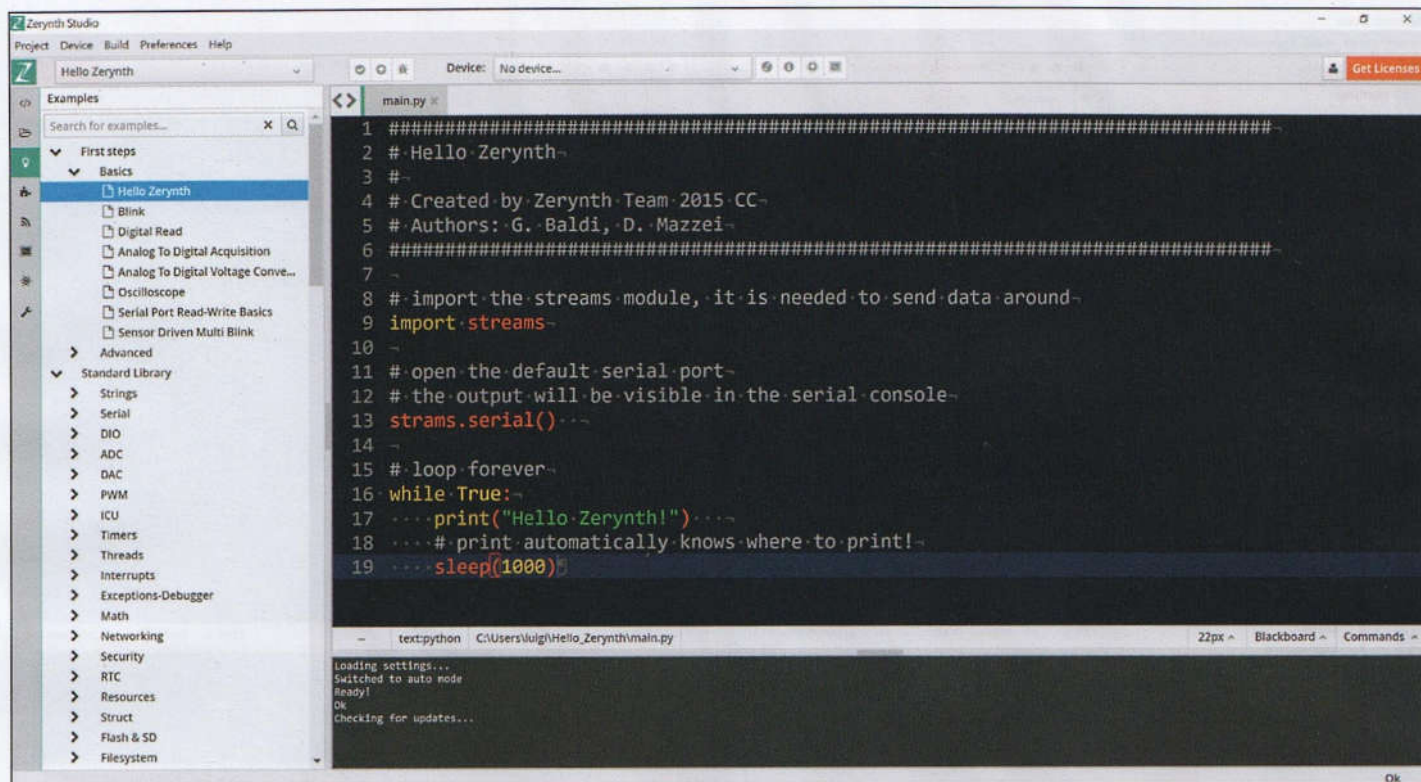


Fig. 2 - Apertura dell'esempio applicativo Hello Zerynth.

indicati dalla freccia rossa.

Cliccando su tali pulsanti possiamo verificare (pulsante Verify) e caricare (pulsante upload) ossia effettuare l'upload dello script nella board. Consigliamo di verificare il progetto prima di caricarlo, facendo clic sul pulsante Verify, così da testare lo script per cercare eventuali errori.

Gli errori rilevati verranno riportati nella console e nel Code Editor, che ci fornirà con utili suggerimenti (Fig. 4).

Per eseguire l'upload del progetto verificato (ed eventualmente corretto) nel dispositivo virtualizzato, facciamo clic sull'icona Uplink e seguiamo le istruzioni fornite a video da Zerynth Studio.

Ed ecco compilato il codice corrispondente:

```
# Hello Zerynth
# import the streams module, it is needed to send data
# around import streams
# open the default serial port, the output will be
# visible in the serial console
streams.serial()
# loop forever
while True:
    print("Hello Zerynth!") # print automatically
    knows
    where to print!
    sleep(1000)
```

pronto per essere caricato nel microcontrollore.
Se volete saperne di più sulla programmazione del-

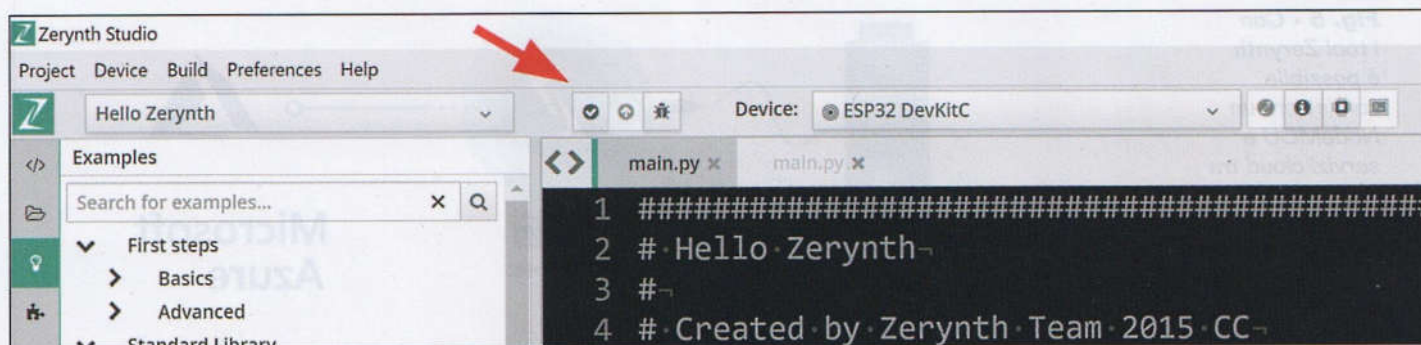


Fig. 3 - I pulsanti per verificare e caricare lo script nel Node MCU.

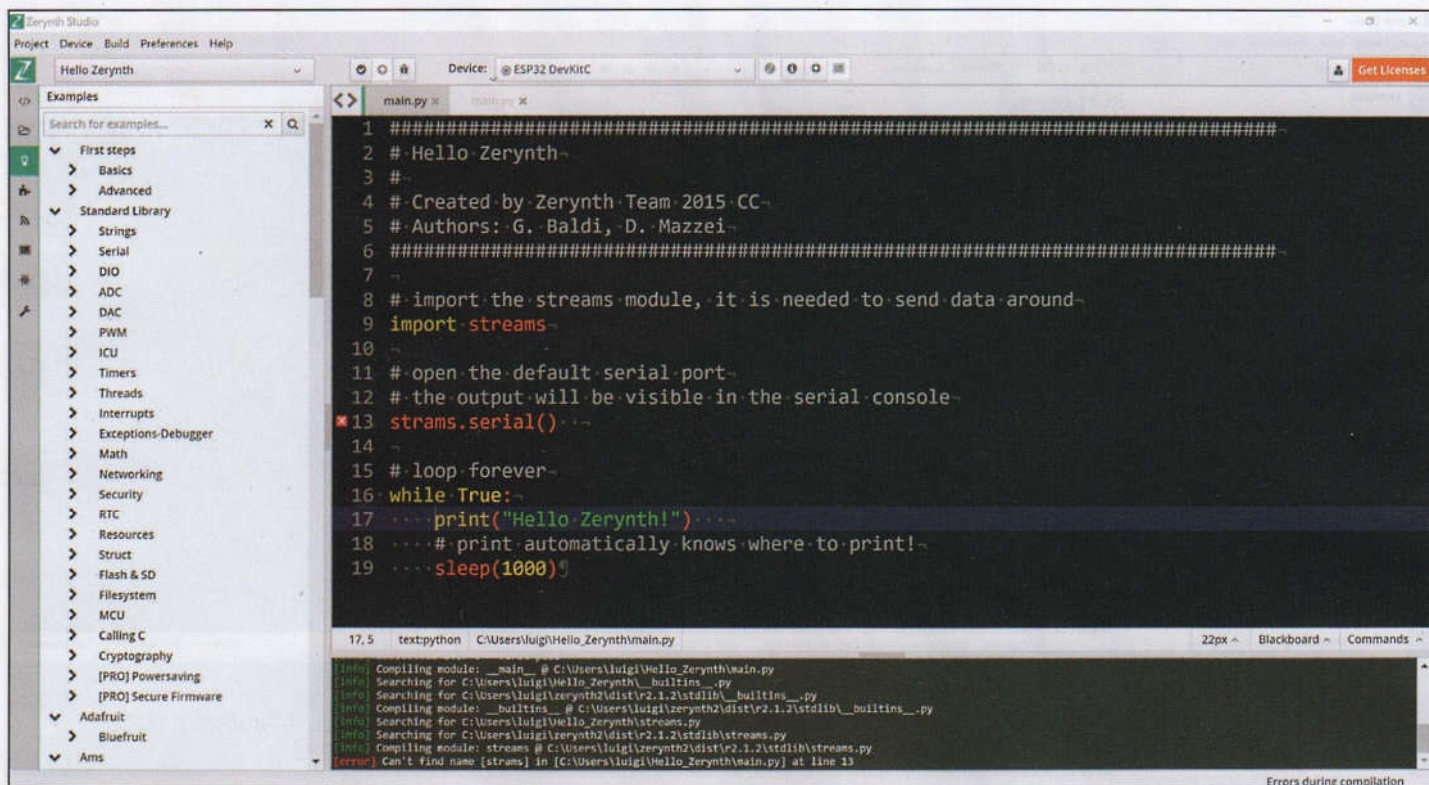


Fig. 4 - Finestra del Code Editor.

la board basata su ESP32 attraverso Python, potete consultare i numerosi video tutorial disponibili sul canale YouTube di Zerynth.

Il flusso di lavoro è lo stesso di quello spiegato poco fa, il che vi renderà facile seguire gli esempi proposti nei video tutorial. Il primo esempio riguarda l'Hello World ed è: **Python on ESP32 DevKitC using Zerynth Studio - Hello World** (<https://www.youtube.com/watch?v=nfx-hJNxdp8>).

Altri video interessanti sono:

- **Python on ESP32 DevKitC using Zerynth Studio**

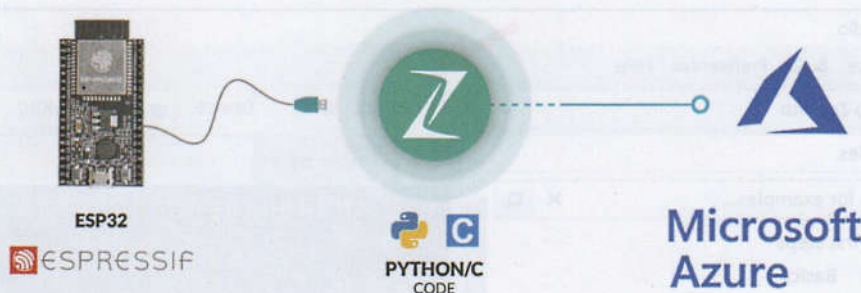
- Mini Web Server (<https://www.youtube.com/watch?v=6Tci77kAMgQ>);

- **Multithreading on Sparkfun ESP32 Thing using Zerynth - Python per Microcontrollers and IoT** (<https://www.youtube.com/watch?v=ZuDbATS2KIA>);

- **Python on DFRobot Firebeetle ESP32 using Zerynth** (<https://www.youtube.com/watch?v=EcVGSPHfYFc>);

- **Read and Write on Serial Port with Zerynth - Python for Microcontrollers and IoT** (<https://www.youtube.com/watch?v=4RT0X0CkU48>).

Fig. 5 - Con i tool Zerynth è possibile connettere un NodeMCU a servizi cloud tra i più importanti come Microsoft Azure.



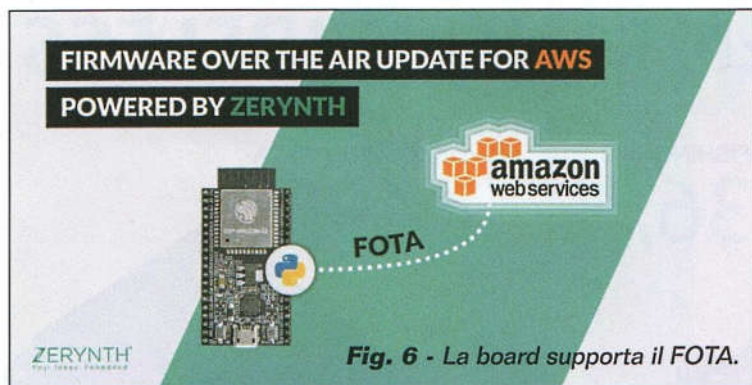


Fig. 6 - La board supporta il FOTA.

CONNETTERE FACILMENTE I MICROCONTROLLORI AL CLOUD

Con Zerynth Studio, connettere il vostro progetto al cloud risulta facile, perché bastano poche righe di codice in Python; infatti nelle intenzioni nel team di sviluppo di Zerynth c'era e c'è l'essere sicuri che questo passaggio nel processo di sviluppo non sia ostico, ma un compito semplice e diretto, scevro da ostacoli insormontabili.

Zerynth vi permette di imparare subito come connettere il NodeMCU dispositivo ad alcuni dei principali fornitori di servizi cloud, come Microsoft Azure, Coogole IoT, Amazon Web Services ed altri ancora. Purtroppo non è scopo di questo articolo, visto che lo spazio richiesto sarebbe eccessivo, spiegare come avviene tale connessione, quindi vi invitiamo a dare un'occhiata ai tutorial qui elencati, accessibili dal sito Internet di Zerynth:

- Connecting IoT devices to Microsoft Azure with Zerynth (<https://www.zerynth.com/blog/connecting-iot-devices-to-microsoft-azure-with-zerynth/>);
- Connecting ESP32 to Google IoT Core with Zerynth Studio (<https://www.zerynth.com/blog/another-great-tutorial-on-thingforward-connecting-esp32-to-google-iot-core-with-zerynth-studio/>);
- Firmware Over-the-air updates via AWS powered by Zerynth (<https://www.zerynth.com/blog/firmware-over-the-air-updates-via-aws-powered-by-zerynth/>).

PREPARATEVI PER INDUSTRY 4.0

Per sfruttare le potenzialità della programmazione in Python per applicazioni industriali, TOI Srl (www.thingsoninternet.it), partner di Zerynth, ha realizzato 4ZeroBox, provvisto di vari GPIO, ingressi e uscite per sensori, slot microSD e connessioni di vario tipo come MikroBUS, CAN, seriale. Se è vero che le applicazioni d'esempio qui proposte rappresentano solo un prototipo, il toolkit industrial 4ZeroBox di TOI si basa proprio sul dispositivo ESP32, quindi grazie a Zerynth Studio è possibile riutilizzare su tale piattaforma il codice

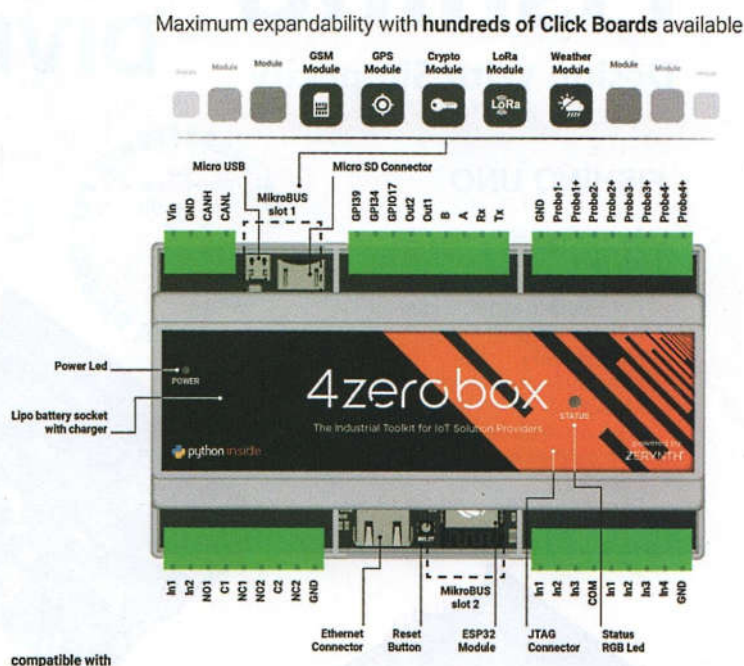


Fig. 7 - La piattaforma hardware 4ZeroBox ti TOI Srl basata su ESP32.

che avete già scritto per il vostro prototipo. Il tutto è consentito dalla Zerynth Virtual Machine e dal livello di astrazione hardware che la caratterizza.

Se volete saperne di più su come lavorare con Zerynth Studio tramite 4zerobox, connettete i dispositivi al cloud, e realizzate applicazioni dell'Industria 4.0 e LoRaWAN, sappiate che Zerynth ha organizzato per il prossimo 11 ottobre a Pisa, The Things Conference on Tour, una conferenza che si terrà presso l'innovativo GATE Center (dove si trovano gli uffici R&D di Zerynth) e che rappresenta una grande opportunità per toccare con mano l'ambiente Zerynth per l'IoT ed entrare in contatto con sviluppatori e progettisti esperti. ■



per il MATERIALE

La scheda di sviluppo e prototipazione basata sull'ESP32 (cod. YB555) è disponibile presso Futura Elettronica al prezzo di Euro 13,50. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - Fax: 0331-792287 - www.futurashop.it

Fishino

Design With Simplicity

I TUOI PROGETTI DIVENTANO WIRELESS

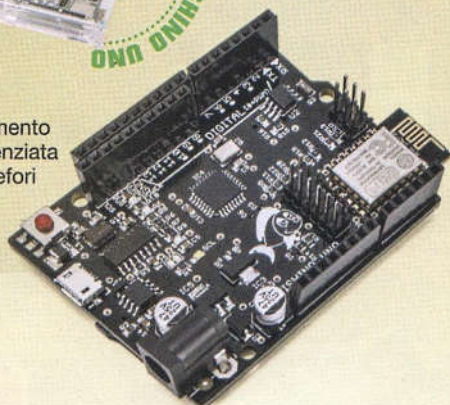
FISHINO UNO

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



cod. FISHINOUNO

€ 36,00



cod. FISHINOMEGA

€ 49,90



FISHINO MEGA

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



FISHINO32

- Controller PIC32 bit
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug o sull'ingresso Vin)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile con Arduino UNO
- Modulo WiFi
- Modulo RTC con batteria di mantenimento
- Clock 120 MHz, riducibile via software
- Interfaccia per scheda microSD
- Codec audio stereo
- Peso 29 g
- Dimensioni 76,5 x 53,5 x 14 mm

cod. FISHINO32

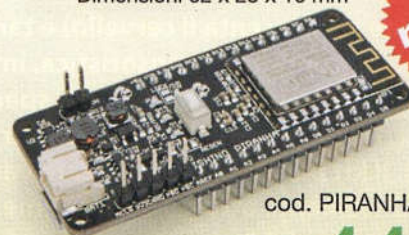
€ 59,00



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

FISHINO PIRANHA

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



cod. PIRANHA

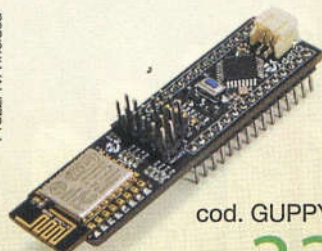
€ 44,90

FISHINO GUPPY

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

cod. GUPPY

€ 33,90



Prezzi IVA inclusa

SHARK, A FISHINO SPUNTANO I DENTI

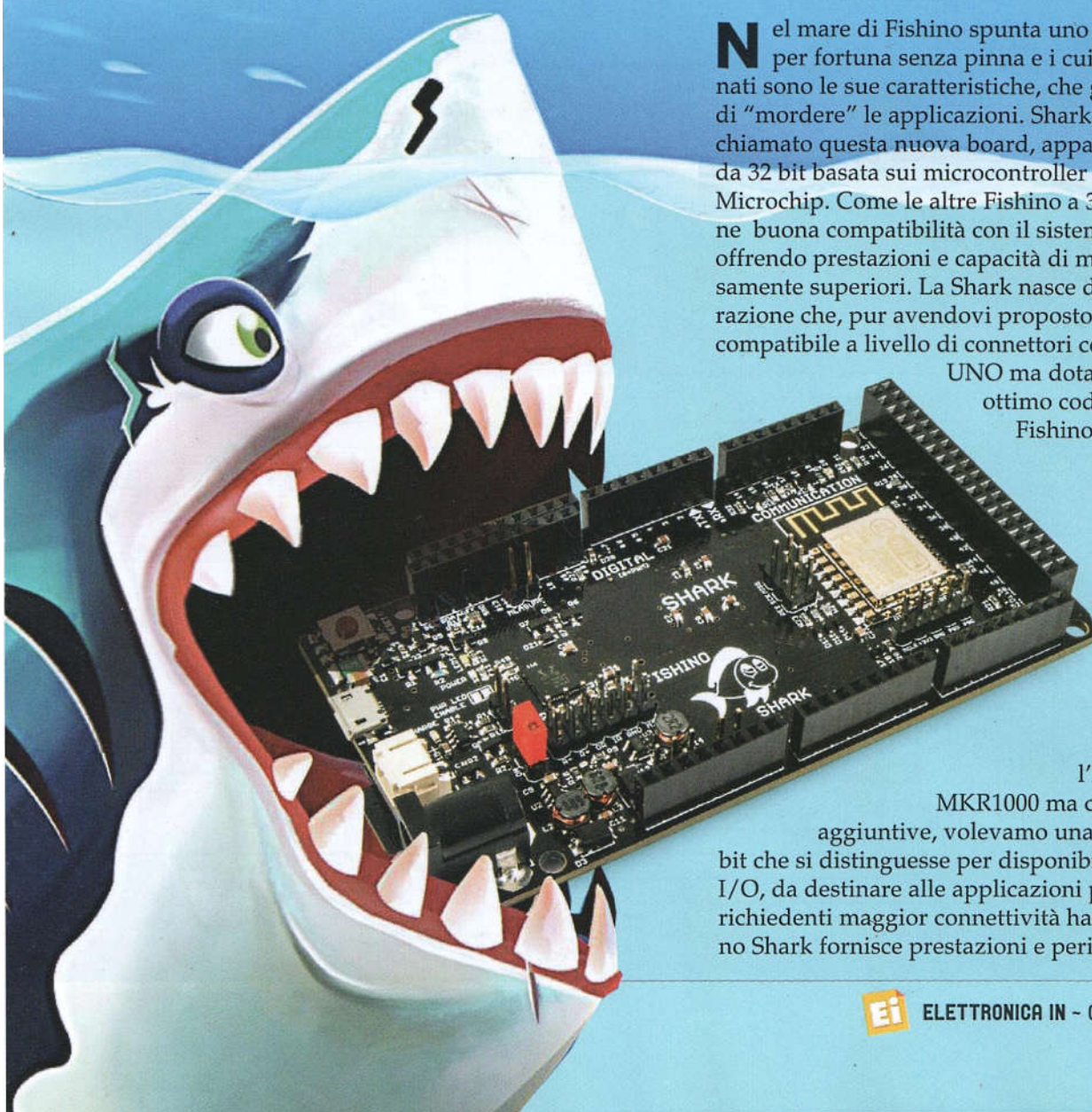
Una nuova scheda Arduino-like appare all'orizzonte ed ha prestazioni tali da poter "divorare" le concorrenti: non a caso è stata chiamata Shark, ossia squalo! Conosciamone le caratteristiche e le novità. Prima puntata.

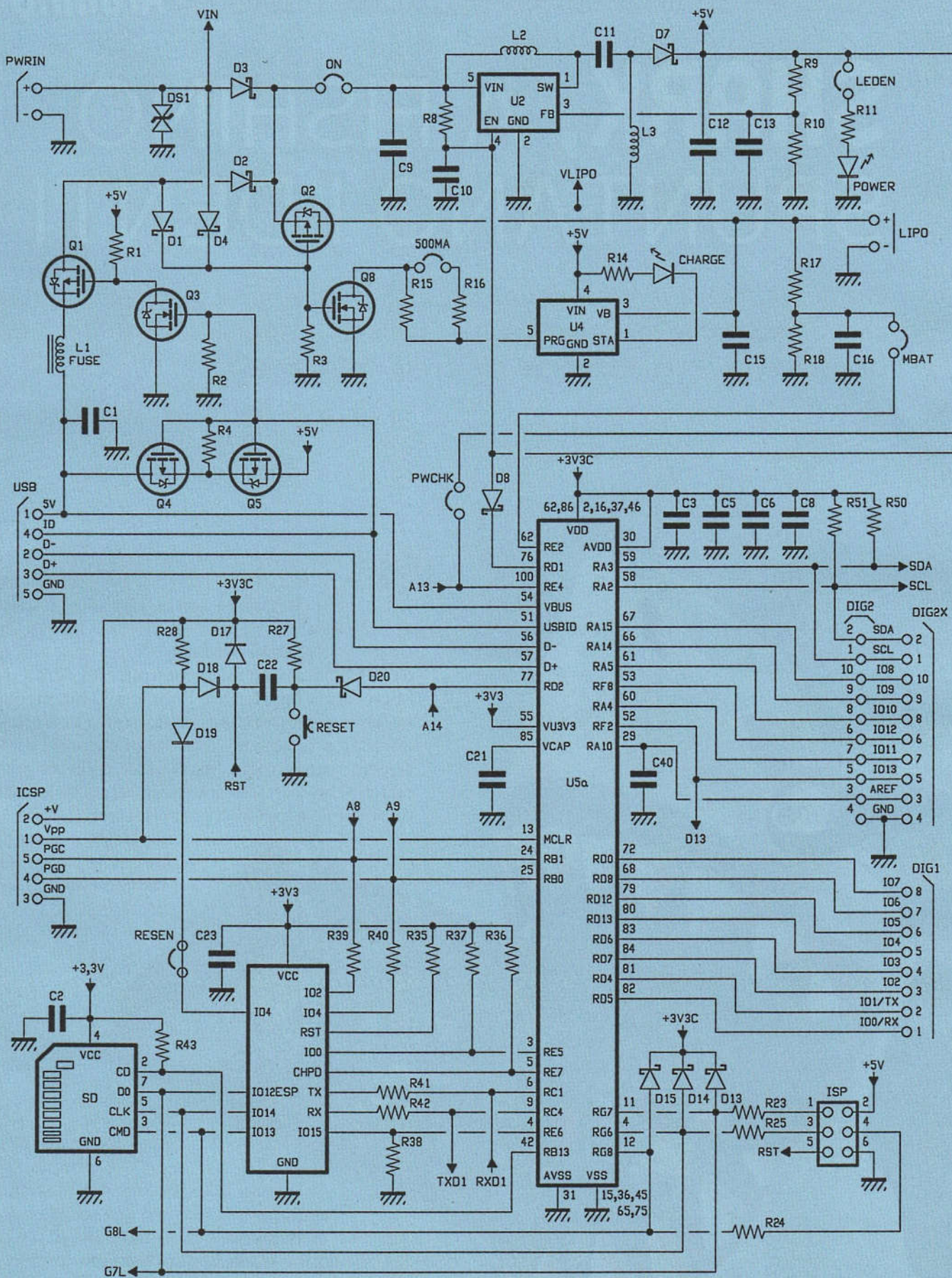
di MASSIMO DEL FEDELE

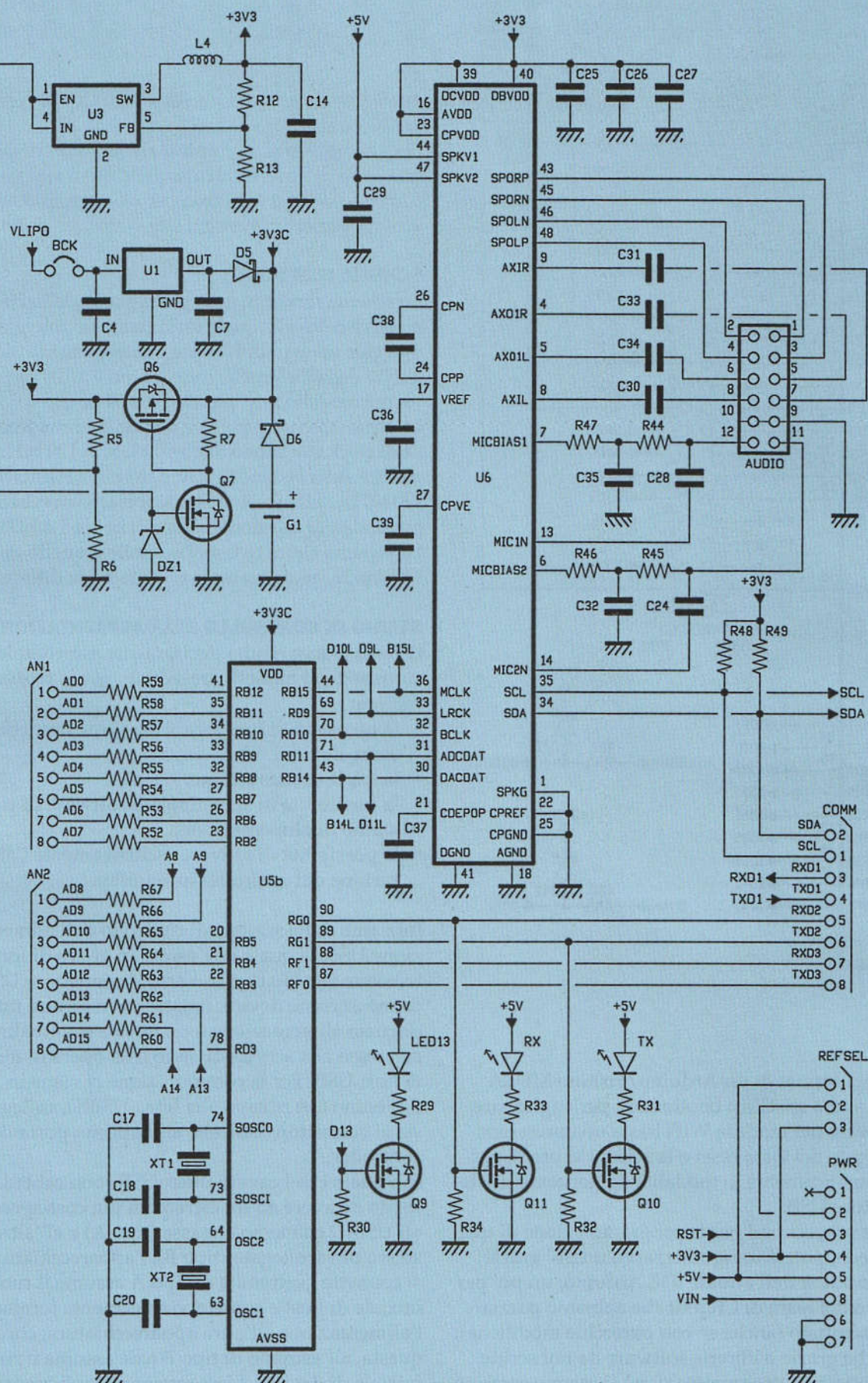
Nel mare di Fishino spunta uno squalo, ma per fortuna senza pinna e i cui denti acuminati sono le sue caratteristiche, che gli consentono di "mordere" le applicazioni. Shark, così abbiamo chiamato questa nuova board, appartiene alla serie da 32 bit basata sui microcontroller PIC32MX della Microchip. Come le altre Fishino a 32 bit, mantiene buona compatibilità con il sistema di Arduino, offrendo prestazioni e capacità di memoria decisamente superiori. La Shark nasce dalla considerazione che, pur avendovi proposto la Fishino32, compatibile a livello di connettori con la Fishino UNO ma dotata in più di un ottimo codec audio, e la Fishino Piranha, com-

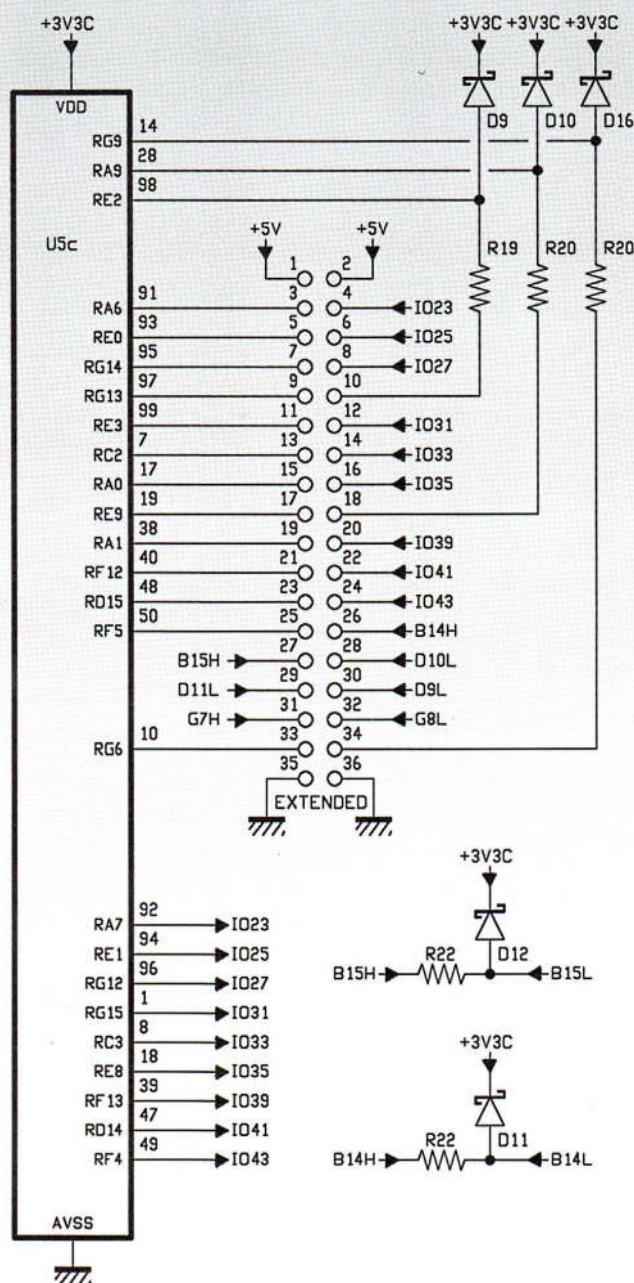
patibile con
l'Arduino

MKR1000 ma con funzionalità aggiuntive, volevamo una scheda a 32 bit che si distinguesse per disponibilità di pin di I/O, da destinare alle applicazioni più complesse e richiedenti maggior connettività hardware. Fishino Shark fornisce prestazioni e periferiche di una









32 bit nel format di un Arduino/Fishino MEGA. Grazie a uno specifico bootloader, per aggiornare il firmware del modulo WiFi basta una pressione prolungata del tasto reset e la scheda si predisporrà automaticamente in modalità aggiornamento da connettore USB.

Se siete preoccupati per la programmazione di questa scheda, potete tranquillizzarvi; un po' grazie alla versatilità dell'attuale IDE Arduino, un po' per il lavoro del team di Chipkit che abbiamo parzialmente sfruttato (anche se con parecchie modifiche), ma anche grazie a librerie software da noi scritte e/o adattate, la programmazione si esegue come su una comune Arduino UNO, ossia tramite l'IDE, nel

quale dovete attivare gli strumenti di sviluppo per le PIC ed il gioco è fatto!

Per i componenti aggiuntivi abbiamo predisposto una serie di librerie, alcune delle quali sono in continuo sviluppo, ragion per cui consigliamo di visitare periodicamente il sito web www.fishino.it.

SCHEMA ELETTRICO

Lo schema ricalca in gran parte quello della Fishino32, rispetto alla quale utilizziamo un micro con maggior numero di I/O che è pressoché identico a quello della Fishino32, solo che qui è a 100 pin mentre nel modello più "piccolo" è a 64 pin.

Lo stesso microcontroller è disponibile con frequenze di clock che vanno dagli 80 MHz ai 120 MHz, però mentre sulla Fishino32 siamo riusciti a sfruttarlo a 120 MHz, sulla Fishino Shark abbiamo previsto la possibilità di selezionare la velocità tra 5 e 105MHz. Lo schema elettrico è molto simile a quello della Fishino32, analizziamolo e vediamo le differenze.

STADIO DI CONTROLLO DELL'ALIMENTAZIONE

Questo blocco risulta decisamente più complesso rispetto ai modelli precedenti, per le seguenti ragioni:

- la necessità di funzionare sia come host che come device USB;
- la tripla alimentazione;
- la possibilità di spegnimento software di tutto lo stadio di alimentazione;
- la possibilità di attivare esclusivamente l'alimentazione del controller in standby.

Iniziamo dalla sezione di controllo dell'alimentazione USB, la quale dev'essere in grado di fornire tensione alla Shark attraverso il connettore USB se connessi come device, mentre deve fornire un'adeguata alimentazione a un device eventualmente connesso alla scheda quando essa opera in modalità host USB. Per la commutazione ci viene in aiuto il piedino che fa capo alla linea USBID, collegata sia al connettore USB che all'apposita porta del controller.

Ricordate che i cavetti microUSB sono cablati in modo da avere ad un estremo il pin corrispondente all'USBID connesso a massa (tipo A) e all'altro lo stesso pin scollegato (tipo B); l'apparecchiatura cui si connette l'estremità di tipo A assume il ruolo iniziale di host e contemporaneamente fornisce l'alimentazione all'altra apparecchiatura, connessa, questa, all'estremo di tipo B (che assume il ruolo iniziale di device). L'espressione "ruoli iniziali" è d'obbligo perché in una connessione OTG, i due

dispositivi USB possono cambiare ruolo; la specifica garantisce soltanto che la linea verrà alimentata sempre dal device connesso all'estremità A.

Torniamo quindi al nostro schema; vediamo che alla linea USBID è connessa una resistenza di valore elevato (R2, 100 KOhm) verso massa, il cui scopo è solo non lasciare fluttuante la linea in caso di assenza di alimentazione, ed una resistenza (R4) di valore decisamente inferiore (10 KOhm) connessa tra i source dei MOSFET Q4 e Q5. Su questi, grazie ai diodi interni ai MOSFET, è sempre presente una tensione positiva (5V meno la caduta diretta sui diodi) proveniente o dalla presa USB o dall'alimentazione a 5V della scheda Fishino. Abbiamo quindi la certezza che la R4 sia connessa ai +5V in qualsiasi condizione di funzionamento. Se la linea USBID è scollegata da massa (connessione tipo B) questa tensione positiva, attraverso R4, porta in conduzione il MOSFET a canale N Q3, il quale a sua volta porta in conduzione il Q1; contemporaneamente, sempre attraverso R4 i due MOSFET Q4 e Q5, vengono posti in interdizione ed essendo collegati in antiserie, non permettono il passaggio dell'alimentazione dai 5V di Fishino al connettore USB.

Sintetizzando, a USBID fluttuante, quindi quando inseriamo un connettore di tipo B, Fishino non fornisce alimentazione al connettore USB che, per contro, alimenta Fishino attraverso il MOSFET Q1. Otteniamo quindi la prima parte dello scopo desiderato, ovvero che con un connettore di tipo B Fishino ricava l'alimentazione dalla linea USB comportandosi da device vero e proprio. Se invece inseriamo un connettore di tipo A, la linea USBID viene posta a massa, cosa che porta in conduzione Q4 e Q5 e contemporaneamente, sempre attraverso Q3, in interdizione Q1; l'alimentazione fluisce quindi da Fishino al connettore USB e non rientra "dalla porta di servizio" essendo Q1 interdetto.

Notate che nel secondo caso (connessione di tipo A) per poter fornire i 5V alla porta USB, Fishino deve essere alimentata dalla batteria o dal plug VIN, come vedremo in seguito.

Proseguendo l'analisi, notiamo le serie di diodi D1+D4 e D2+D3: i primi due sono per piccole correnti, mentre i secondi sono di potenza. Vediamo inoltre il MOSFET Q2, al quale giunge l'alimentazione della batteria. I diodi D2 e D3, insieme al MOSFET Q1, costituiscono una porta OR di potenza che confluisce nella linea VOR dello schema. Per i diodi, il funzionamento è questo: la tensione maggiore tra VIN e VUSB viene trasmessa su VOR, mentre l'altra, a causa dell'interdizione del diodo corrispondente, che si trova polarizzato inversa-

mente, viene isolata.

Ci si potrebbe chiedere perché sulla linea della batteria non abbiamo inserito semplicemente un diodo di potenza come per le altre due, cosa che effettivamente avrebbe semplificato il circuito, realizzando una porta OR di potenza con tre diodi. Questo è fattibile, ma resta il problema della caduta di tensione sui diodi che, se per tensioni elevate è praticamente influente, su una tensione ridotta come quella della batteria porta a perdite considerevoli: si può dire che una caduta di tensione di 0,5V su 10V corrisponde al 5%, mentre la stessa caduta su 3,6V incide per circa il 14%, cosa che si riflette sull'assorbimento in corrente come vedremo di seguito, e proprio dove servirebbe una maggior efficienza, ovvero nell'alimentazione a batteria. Il MOSFET Q2 non ha questo problema, perché diversamente dal diodo non presenta una caduta di tensione minima ma una resistenza di conduzione, molto bassa nel modello utilizzato; per contro, un MOSFET non ha il pregio del diodo di condurre solo quando serve, per cui occorre un sistema per mandarlo in conduzione o interdirlo. Per questo utilizziamo i due diodi per piccole correnti D1 e D4; il funzionamento è il seguente:

- se non è presente un'alimentazione su VIN e nemmeno attraverso VUSB, la resistenza R3 porta a massa il gate di Q1 che quindi conduce e permette il passaggio della tensione della batteria verso VOR;

CARATTERISTICHE TECNICHE

- Formato standard Arduino MEGA
- Processore PIC32MX470F512
- 512 kB di ROM
- 128 kB di RAM
- Frequenza di clock di 105 MHz
- Interfaccia USB nativa, sia device che host
- RTC incorporato
- Lettore per schede microSD
- Modulo WiFi integrato
- Codec audio
- Tensione di alimentazione: 3÷20Vcc
- Alimentazione a batteria, plug e/o via connettore USB
- Funzionamento interno a 3,3 V
- Pin DIGITALI 5V-tolerant
- Ricarica automatica di una batteria LiPo
- Spegnimento da software con processore in standby
- Wake-up tramite pin esterno o in tempi prefissati

- se invece è presente una delle due (VIN o VUSB, o entrambe) uno dei diodi D1 e D4 conduce, polarizzando positivamente Q2 che passa in interdizione scollegando quindi la batteria.

In breve, se abbiamo un'alimentazione esterna, che proviene dal plug VIN o dal connettore USB, la batteria viene scollegata; se invece manca l'alimentazione esterna entra in funzione la batteria.

Si può notare che la linea su cui confluiscono i diodi D1 e D4 ha un nome, BATOFF, perché questo segnale, che è a livello alto se è presente un'alimentazione esterna, verrà utilizzato anche in un'altra sezione del circuito.

Passiamo alla sezione circuitale facente capo ai MOSFET Q6 e Q7; questi servono per scollegare la linea +3V3, ovvero la linea principale a +3,3V, dalla linea +3V3Core, destinata ad alimentare solo il controller. A cosa servono? Semplicemente ad evitare che un'alimentazione destinata al solo microcontrollore, attivata quando tutto il resto è spento allo scopo di mantenere attivo l'RTC interno ed altre funzionalità minime, confluisca nei circuiti che intendiamo tenere spenti. Normalmente, se è presente tensione sui +3,3V, questa confluisce attraverso il diodo interno a Q6, e anche attraverso il Q6 stesso se questo è in conduzione, sull'alimentazione +3V3Core del controller; è questo il funzionamento normale. Invece se togliamo l'alimentazione sui +3,3V, e contemporaneamente portiamo a livello basso la linea OFF, attraverso Q7 portiamo il Q6 ad interdizione e quindi impediamo ai +3V3Core di andare ad alimentare il resto della scheda. La linea OFF viene comandata dal controller, come vedremo in seguito, durante lo spegnimento programmato.

Si possono poi notare la pila al litio di mantenimento G1 ed il diodo D6, che servono a fornire

tensione al controller quando manca qualsiasi altra cosa in grado di alimentarlo. Per questo occorre fare attenzione: se togliamo completamente l'alimentazione al Fishino, questa piletta finisce per alimentare tutta la scheda, a meno di non "prendere provvedimenti" attraverso il software portando a livello basso la linea OFF. A differenza quindi delle schede ad 8 bit, dove la piletta serviva solo per la tensione di mantenimento dell'RTC, e la cosa veniva gestita in modo automatico, qui è necessario un intervento del software a supporto, altrimenti la piletta si scaricherebbe in pochi minuti.

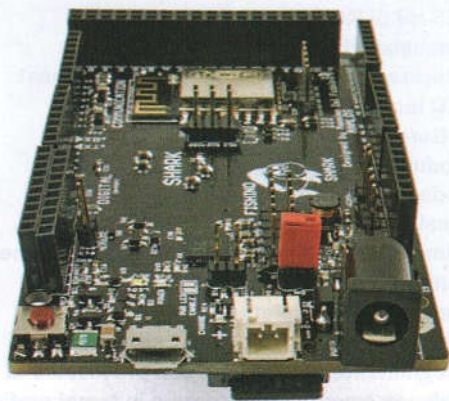
Consigliamo quindi di non utilizzarla, e lasciare il mantenimento alla batteria LiPo ricaricabile, a meno che non si sappia come gestire il risparmio energetico della scheda via software.

Per ultima notiamo la linea PWRGOOD, connessa ad un partitore con la massa; questa fornisce un segnale positivo ad un I/O del controller che indica la presenza di un'alimentazione adeguata sulla scheda, e può venire utilizzato, appunto, per gestire la mancanza di questa e le funzioni di stand-by. Come detto sopra si tratta di una gestione piuttosto complessa, per la quale è stata predisposta un'apposita libreria software, tuttora in fase di sviluppo, per gestire le modalità di stand-by. Vi consigliamo di guardare il codice e gli esempi della libreria FishinoLowPower per capirne il funzionamento!

L'ultima parte della sezione in esame è quella che ruota attorno al regolatore lineare U1, un semplice regolatore a bassa corrente (max 100 mA circa) e bassissimo consumo, destinato a fornire la tensione di mantenimento al controller tramite la batteria LiPo esterna.

Questo ha lo stesso effetto della piletta al litio G1, con l'indubbio vantaggio di utilizzare una batteria ricaricabile in grado di fornire una durata estremamente lunga anche in presenza di consumi non proprio trascurabili.

Questa sezione è disattivabile tagliando il ponticello BCK, normalmente connesso. Anche qui valgono (anche se in modo meno pressante) i discorsi sul risparmio energetico fatti prima: se spegniamo via software la scheda, il controller prenderà alimentazione dalla batteria LiPo che quindi tenderà a scaricarsi; inoltre, se superassimo i 100 mA di assorbimento, il piccolo regolatore lineare si "siederebbe", annullando la sua tensione in uscita. È quindi sempre compito del software fare in modo che il controller venga posto in standby o comunque in una delle modalità a basso consumo disponibili.



STADIO DI ALIMENTAZIONE SWITCHING

Questo stadio fa capo agli integrati U2 (SX1308), un convertitore asincrono step-up qui utilizzato in modalità SEPIC in grado di fornire un'uscita a 5V con tensioni in ingresso variabili da circa 3 a 20 V, ed all'integrato U3, che è un convertitore sincrono step-down, utilizzato per ricavare i 3,3 V di alimentazione per la maggior parte del circuito.

Entrambi questi stadi sono già stati ampiamente descritti per le schede Fishino Mega e Guppy e, per quanto riguarda il funzionamento del SEPIC, nella scheda Torpedo. Le uniche differenze consistono nella presenza del jumper ON, che permette l'inserimento di un interruttore di spegnimento in grado di togliere l'alimentazione a tutta la scheda (fatto salvo quella di mantenimento eventualmente abilitata, come spiegato nel paragrafo precedente), e la linea OFF, già vista prima, in grado di spegnere, se portata a massa, il convertitore switching.

Questa linea è sotto diretto controllo del processore, attraverso un I/O e permette quindi sia lo spegnimento completo della scheda via software, sia l'isolamento della sezione di alimentazione del solo controller dal resto del circuito.

Riassumendo, ponendo la linea OFF a massa viene tolta tutta l'alimentazione alla scheda; ovviamente questa linea deve essere mantenuta a massa dal controller che quindi deve rimanere operativo attraverso una delle due sorgenti di mantenimento (LiPo o piletta al litio); in assenza di queste, portando la linea OFF a massa otterremo solo un reset della scheda ed il conseguente riavvio del tutto.

STADIO CARICABATTERIA

Anche questo stadio, come nelle board precedenti, fa capo all'integrato MCP73831, un completo regolatore di carica integrato per LiPo a cella singola. Qui, l'unico elemento degno di nota è il mosfet Q8, un canale N utilizzato per abilitare o meno la funzione di carica della batteria.

Se ricordiamo la descrizione del primo stadio (controllo alimentazioni), la linea BATOFF è a livello alto quando è presente un'alimentazione esterna; in questo caso il mosfet Q8 va in conduzione, portando a massa la resistenza R15 (ed eventualmente la R16 se abbiamo selezionato la carica a 500 mA) attivando quindi la carica. In assenza di tensioni esterne (alimentazione a batteria) la linea BATOFF va a livello basso disattivando la carica.

A completamento della descrizione notiamo le resistenze R17 ed R18, che costituiscono un partitore

di tensione in grado di fornire metà della tensione della batteria LiPo, misurabile attraverso una linea del controller per poter segnalare lo stato di carica della stessa. Abbiamo utilizzato un partitore perché la tensione della batteria LiPo, a piena carica, supera i 3,3 V tollerati dagli ingressi analogici del microcontrollore; occorre tenerne quindi conto nella misura.

SEZIONE MICROCONTROLLORE

Arriviamo finalmente al PIC32MX470F512L, un microcontroller a 32 bit capace di elevate prestazioni, consumi decisamente ridotti e con una ricca dotazione di periferiche.

Notiamo innanzitutto due quarzi, uno da 20 MHz e l'altro da 32,768 KHz. Ma perché due quarzi? Ebbene, perché il primo fornisce il clock principale, attraverso un PLL interno che lo moltiplica fino a 105 MHz; si tratta di un circuito non proprio a basso consumo, quindi da evitare nelle condizioni di risparmio energetico. Il quarzo da 32.768 Hz serve esclusivamente per il real-time-clock (RTC) interno al PIC, che è un circuito studiato appositamente per avere un bassissimo assorbimento e che può quindi essere lasciato in funzione anche in condizioni di stand-by.

Tornando al primo quarzo, accettando una precisione minore sulla frequenza questo avrebbe potuto essere rimosso perché il PIC integra un oscillatore RC piuttosto preciso, ma che ha due limiti:

- non fornisce la precisione del quarzo;
- non è in grado di portare il PIC alla sua massima frequenza di 105 MHz, a causa di vincoli del PLL interno che limitano a 96 MHz.

Il primo punto ha importanza solo se utilizziamo la scheda per misure precise di tempi brevi: ad esempio come frequenzimetro o periodometro; per i tempi lunghi basterebbe l'altro quarzo.

Il secondo punto è vincolante se vogliamo "spremere" il controller al massimo delle sue possibilità. Del micro notate le già accennate linee OFF, che gli permette di togliere l'alimentazione principale, PWRGOOD, che consente il controllo dello stato della medesima, ed MBAT che permette la misura della tensione della batteria.

In questa scheda, visto che gli I/O accessibili esternamente sono parecchi, abbiamo dovuto "multiplexare", ovvero sovrapporre, alcune funzionalità su singoli pin: la PWRGOOD, ad esempio, è sovrapposta all'ingresso analogico AD13; se si utilizza quest'ultimo la funzionalità di PWRGOOD non è disponibile e viceversa.

STADIO RESET E LED

Questo stadio è piuttosto interessante perché ci ha permesso di utilizzare un solo pulsante di reset per svariate funzioni, in abbinamento con un apposito bootloader sul controller.

Notiamo subito tre linee che fanno capo allo stadio di reset:

- MCLR
- REMOTERESET
- LONGRESET

La linea MCLR è il reset vero e proprio e portando a massa il microcontrollore viene riavviato. Notiamo subito che questo può avvenire sia premendo il pulsante di reset (attraverso il condensatore C22 che trasferisce un breve impulso a zero logico alla linea MCLR), sia portando a massa la linea REMOTERESET, proveniente dal modulo WiFi e che consente quindi a questo di resettare il micro.

La linea LONGRESET, per contro, viene a trovarsi a massa per tutto il tempo in cui il pulsante di reset viene mantenuto premuto, consentendo quindi al bootloader di conoscere le nostre intenzioni.

In sintesi:

- una breve pressione del pulsante reset (inferiore ai 2 secondi) o uno zero logico sulla linea REMOTERESET riavviano la scheda;
- mantenendo premuto il pulsante di reset per un tempo tra 2 e 4 secondi viene avviato il bootloader (cosa evidenziata dal lampeggio del LED arancione), mettendo quindi la scheda in modalità caricamento sketch;
- mantenendo premuto il pulsante per un tempo superiore ai 4 secondi si avvia la procedura di upgrade del firmware del modulo WiFi, cosa evidenziata dal lampeggio del LED blu.

È comunque possibile, una volta entrati nelle due modalità del bootloader, riavviare semplicemente la scheda ed eseguire lo sketch precaricato premendo nuovamente e brevemente il pulsante di reset. Anche qui, per scarsità di pin, abbiamo dovuto condividere la linea LONGRESET con un I/O analogico (l'AD14); nessun problema, in questo caso, visto che premendo il reset l'ingresso non serve!

Al microcontrollore abbiamo assegnato due LED (pilotati attraverso i MOSFET Q10 e Q11 per non caricare eccessivamente gli I/O del PIC), multiplexati con la porta seriale 2, utilizzabili per qualsiasi compito (indicati sulla scheda come RX/PRG e TX, rispettivamente); utilizzando la porta seriale 2, i LED pulseranno in analogia con i dati in ingresso/uscita. Se non usiamo la seriale 2, possiamo gestire

i LED via software.

Al microcontrollore troviamo inoltre collegato il solito LED connesso alla porta 13 che viene utilizzato nei test della scheda (programma BLINK), attraverso il mosfet Q9 per non caricare l'uscita D13. A differenza della Fishino32, l'uscita D13 è un normale I/O e NON ha la funzionalità aggiuntiva di clock per l'interfaccia SPI, che è, come per l'Arduino/Fishino MEGA, su altri pin.

STADIO WIFI

Anche questo stadio è già stato descritto nelle precedenti schede, ma qui troviamo una differenza piuttosto consistente, quindi lo rivediamo in dettaglio.

Il modulo sfrutta sempre un firmware da noi realizzato, in grado di comunicare ad alta velocità tramite la porta SPI e non sulla seriale, come d'abitudine nei sistemi che utilizzano l'ESP8266; la differenza di prestazioni, come già notato, è notevole.

In questa scheda, per contro, non vediamo più il connettore ESPCONN al quale eravamo abituati nelle board precedenti ad 8 bit ed al quale facevano capo alcuni I/O e pin di controllo del modulo sfruttati, sia per programmarlo che, per alcuni di essi, come I/O aggiuntivi.

Come mai questa scelta? Semplicemente perché nella Fishino Shark abbiamo già liberato alcuni degli I/O che venivano usati per comunicare con il modulo e con la scheda SD, che erano il 4, 7 e 10; abbiamo inoltre separato la porta I²C dai due ingressi analogici A4 ed A5, ottenendo quindi 3 porte digitali e due porte analogiche in più a disposizione. Non solo, ma due porte sul connettore ICSP sono anch'esse sfruttabili come porte digitali aggiuntive (ne parleremo in dettaglio descrivendo il software), quindi abbiamo preferito sfruttare le connessioni con il modulo ESP in altro modo.

Infatti, quasi tutte le linee dell'ESP (salvo l'ingresso analogico) sono ora sotto controllo del processore principale, cosa che ci permette di effettuare le seguenti operazioni direttamente da software, senza impegnare alcun I/O:

- resettare il modulo da software;
- spegnere il modulo da software (cosa che in precedenza richiedeva l'uso di un I/O del Fishino);
- caricare il firmware sul modulo senza dover effettuare collegamenti strani né caricare sketch particolari;
- ridirigere l'output seriale del modulo sulla seriale USB o su qualsiasi altra seriale, sia software che hardware, disponibile sul Fishino.

Ma perché "quasi" tutte le linee? Anche qui, per risparmiare un I/O sul controller, abbiamo evitato di collegare la linea di RESET dell' ESP8266; la stessa funzionalità si può ottenere tranquillamente tramite la linea ESP_CHPD, che serve anche a mandare in modalità low-power il modulo.

Le nostre librerie software tengono già in conto tutto questo.

Per il resto, cambia praticamente nulla rispetto alle schede precedenti: la comunicazione avviene attraverso la porta SPI, che fa capo anche alla scheda SD ed ai pins digitali D50, D51 e D53 sul connettore grande in fondo alla scheda.

Visto che la PIC utilizzata ha due interfacce SPI incorporate, verrebbe da chiedersi perché non è stata dedicata una di queste ai moduli inclusi (WiFi e SD) liberando così anche gli I/O D50, D51 e D53 completamente; questo è stato fatto perché la seconda interfaccia SPI è dedicata, in modalità I²S, al codec audio che quando è in funzione la impegna a tempo pieno.

In un primo tempo avevamo scelto quella strada, salvo accorgerci in seguito che risultava impossibile, ad esempio, riprodurre un file audio letto da scheda SD, visto che durante la riproduzione l'interfaccia era totalmente impegnata con il codec senza possibilità di accedere alla SD.

Va detto che nelle schede in formato Arduino MEGA i tre I/O suddetti sono comunque dedicati alla porta SPI, e comunque se necessario abbiamo a disposizione i due I/O sulla porta ICSP di cui parlavamo prima, ed eventualmente tutti gli I/O analogici che sono comunque utilizzabili anche in digitale.

STADIO INTERFACCIA MICROSD

Questo stadio è assolutamente identico a quello delle schede già viste; sfrutta l'interfaccia SPI, come il modulo WiFi, mentre anche qui per la selezione della scheda viene utilizzato un I/O interno, lasciando libero quindi l'I/O D4 che prima era necessario per questo scopo.

L'unica grossa differenza rispetto alle schede precedenti è l'assenza degli stadi di conversione livelli (da 5 Volt a 3,3 V) visto che la PIC utilizza i 3,3 volt come alimentazione, rendendo quindi possibile (e più veloce!) un collegamento diretto. L'unico dettaglio da tenere presente nell'esecuzione di sketch esistenti è che il pin di selezione scheda (CS) non è più il 4 (o il 10 in alcune versioni di Arduino) ma un pin interno denominato SD_CS; è quindi sufficiente sostituire tutti i riferimenti all'I/O 4 con l'I/O SD_CS (scritto proprio così, visto che può cambiare fisicamente da scheda a scheda).

STADIO CODEC AUDIO

Ed eccoci arrivati alla succosa novità presente nel Fishino32, il codec audio!

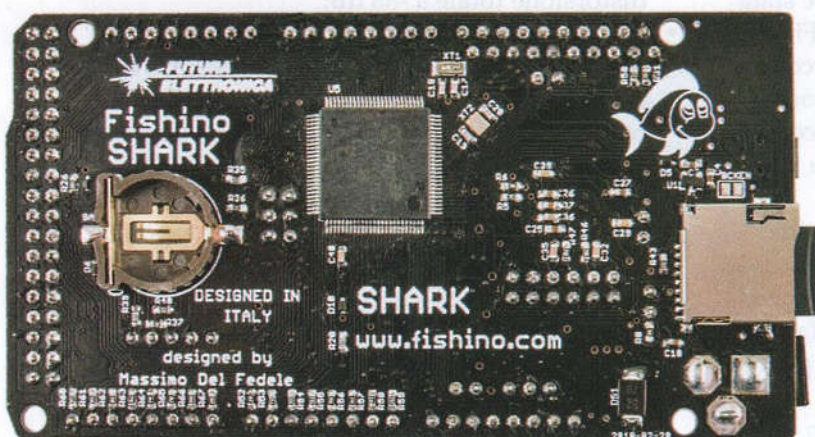
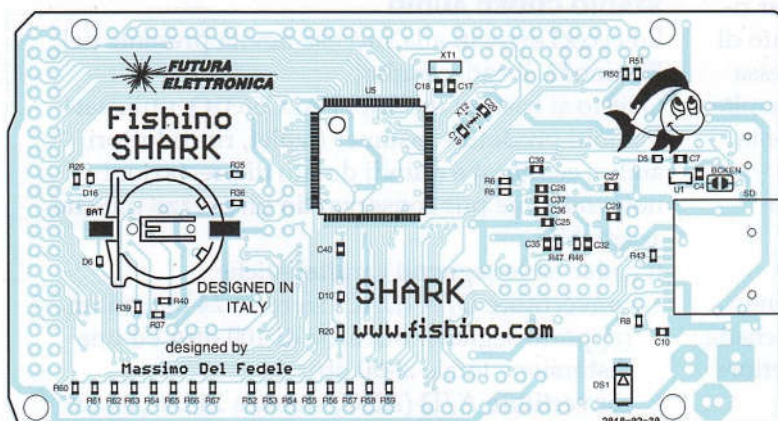
Questo si basa su un chip, l'ALC5631Q, utilizzato in molti prodotti consumer (tablet, riproduttori audio, eccetera) e quindi dotato di prestazioni notevolissime pur conservando un prezzo più che accettabile.

Il codec ha le seguenti caratteristiche:

- convertitore D/A (DAC) stereo a 24 bit con un rapporto segnale/disturbo di 100 dB ed una distorsione totale a -90 dB;
- convertitore A/D (ADC) stereo a 24 bit con un rapporto segnale/disturbo di 93 dB ed una distorsione totale a -88 dB;
- un amplificatore stereo in classe D in grado di erogare 650 mW per canale se alimentato a 3,3V su un'impedenza di 8 ohm e fino a 2,8 W per canale se alimentato (come nel nostro caso) a 5 volt, su un'impedenza di 4 ohm;
- un amplificatore per cuffie stereo, in grado di erogare 45 mW su 16 ohm, che purtroppo non utilizzeremo nella nostra scheda per motivi di spazio;
- 3 ingressi analogici differenziali o un ingresso analogico stereo (noi utilizziamo quest'ultimo);
- 2 ingressi microfonici analogici preamplificati, sia in modalità differenziale (da noi non usata) sia in modalità single-ended;
- un ingresso per microfono stereo digitale (anche questo, non utilizzato sulla nostra scheda sempre per motivi di spazio);
- un'uscita analogica di linea, alla quale è possibile connettere un amplificatore esterno;
- un completo sistema di power management che permette di spegnere i singoli componenti del codec se non utilizzati;
- un equalizzatore digitale a 6 bande audio;
- effetti di pseudo-stereo e di stereo spaziale;
- circuiti di rimozione dei disturbi di accensione e spegnimento negli altoparlanti;
- gestione automatica di livelli (ALC) sia in riproduzione che in registrazione;
- interfaccia standard I²S ad alta velocità (24 bit/ da 8 a 192 KHz sul DAC in riproduzione e 24 bit/ da 8 a 96 KHz sull'ADC in registrazione).

Come si può notare, una serie di caratteristiche impressionanti, che permetteranno di realizzare, senza alcun hardware aggiuntivo che non sia un altoparlante o un microfono, moltissimi progetti che coinvolgono l'audio digitale. Tra questi:

- player audio (wav e Mp3);



Elenco Componenti:

- C1: 100 μ F ceramico
- C2, C3, C5, C6, C8, C10, C16, C40: 100 nF ceramico
- C4, C7, C21, C22, C24, C25, C26, C27, C28, C30, C31, C33, C34, C37: 1 μ F ceramico
- C9: 22 μ F 25 VL ceramico
- C11: 4,7 μ F 25 VL ceramico
- C12: 22 μ F ceramico
- C13: 360 pF ceramico
- C14: 10 μ F ceramico
- C15, C23, C32, C35, C36: 4,7 μ F ceramico
- C17÷C20: 22 pF ceramico
- C29: 22 μ F ceramico
- C38, C39: 2,2 μ F ceramico
- D1: RB521S
- D2, D3: SS34SMA
- D4÷D6, D8÷D20: RB521S
- D7: SS34SMA
- DZ1: MM3Z3V3
- DS1: SMBJ24CA
- Q1: FDN340P
- Q2: NTR4171P
- Q3: 2N7002
- Q4, Q5: FDN340P
- Q6: NTR4171P
- Q7÷Q11: 2N7002
- R1, R2: 100 kohm
- R3÷R5, R15, R28, R35÷R38, R43, R48÷R51: 10 kohm
- R6÷R8: 220 kohm
- R9: 97,6 kohm
- R10: 13,3 kohm

- registratore audio;
- riconoscimento vocale;
- sintesi musicale;
- sintesi vocale.

Come vedrete in seguito, abbiamo già implementato un player per files .wav, .mp3 e .ogg/.ogb; per la registrazione abbiamo a disposizione una libreria in grado di creare files .wav. Esiste anche un encoder MP3, ma le sue prestazioni sono decisamente insufficienti ed è stato incluso solo per curiosità e per poter fare alcune prove; la compressione MP3 è infatti troppo onerosa per essere effettuata in real time dal nostro pur potentissimo controller.

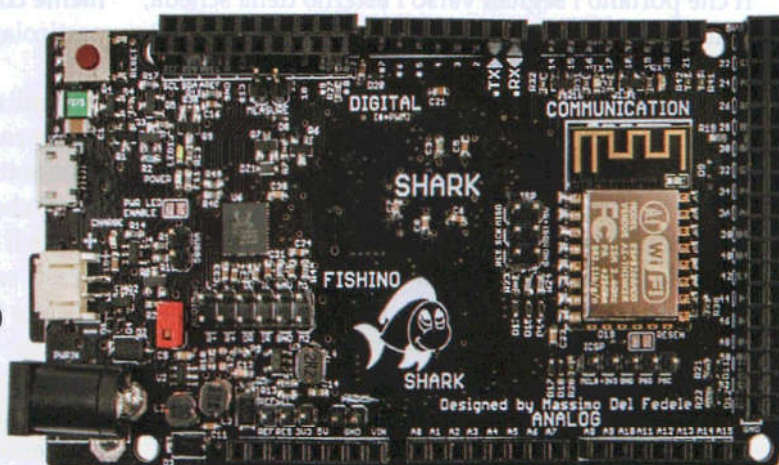
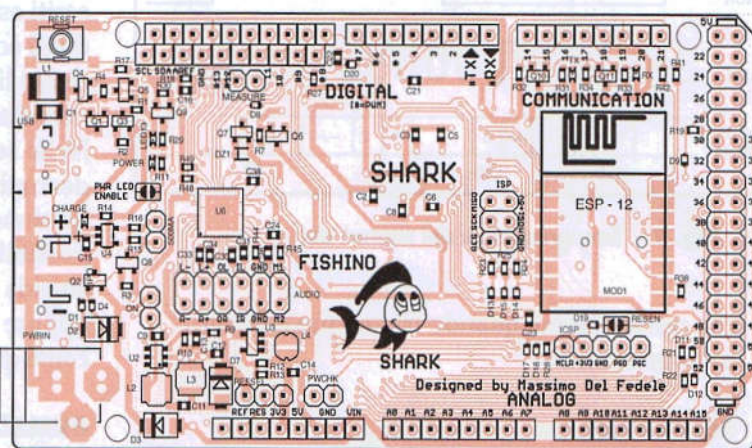
Come già accennato parlando del modulo WiFi, il codec sfrutta un'interfaccia SPI dedicata, utilizzata in modalità I²S, per la gestione della quale abbiamo scritto un'apposita libreria software.

Il codec ha inoltre bisogno di una porta I²C (non necessariamente dedicata solo a lui, ed infatti ab-

biamo utilizzato la porta I²C standard del Fishino) per controllare tutte le varie funzioni disponibili. Anche qui abbiamo creato un'apposita libreria in grado di controllare il funzionamento del codec; tramite questa si può, per esempio, decidere se registrare o riprodurre, modificare il volume audio, regolare l'equalizzatore, impostare l'ALC, eccetera. Come potrete ben immaginare, i controlli sono numerosissimi, quindi la libreria verrà estesa man mano che se ne presenterà la necessità. Sullo schema elettrico della sezione c'è pochissimo da aggiungere: come potete vedere, oltre al codec abbiamo qualche condensatore di disaccoppiamento sulle alimentazioni e sugli I/O, un condensatore per un convertitore interno destinato ad aumentare la tensione per l'amplificatore delle cuffie (C38) ed un paio di resistenze per la tensione di bias (polarizzazione) dei microfoni. Null'altro!

Torniamo ora un momento sull'interfaccia I²S: si tratta di un'interfaccia basata sull'SPI, prevista

R11, R27,	500mA
R41, R42: 1 kohm	L2: 6.8 μ H
R12: 475 kohm	L3: 6.8 μ H
R13: 105 kohm	L4: 2.2 μ H
R14, R29,	USB: Connettore
R31, R33: 470 ohm	micro-USB
R16: 2,7 kohm	SD: Connettore micro-SD
R17, R18: 1 Mohm	MOD1: Modulo ESP12
R19÷R26: 680 ohm	con ESP8266
R30, R32, R34: 220 kohm	LIPO: Connettore JST
R39: 3,3 kohm	2 vie 2.54 mm
R40: 3,3 kohm	BAT: Porta batterie
R44, R45: 680 ohm	Ø12 mm per CR1220
R46, R47: 2,2 kohm	PWRIN: Plug alimentazione
R52÷R67: 220 ohm	RESET: Microswitch
U1: XC6206P33	
U2: SX1308	Varie:
U3: LC3406	- Strip femmina 8 vie
U4: MCP73831	(5 pz.)
U5: PIC32MX470F512L-120I/PF	- Strip femmina
U6: ALC5631Q	10 vie (2 pz.)
XT1: 32.768 KHz	- Strip femmina 2x18 vie
XT2: 20MHz	- Strip maschio 2x3 vie
TX: LED blu	- Strip maschio 2x6 vie
RX/PRG: LED arancione	- Strip maschio 2 vie (4 pz.)
LED13: LED bianco	- Strip maschio 3 vie
POWER: LED verde	- Strip maschio 5 vie
CHARGE: LED rosso	- Jumper (4 pz.)
L1: MF-MSMF050-2	- Circuito stampato S1408



espressamente per la trasmissione di flussi audio stereo ad alta velocità; a differenza della SPI e a causa del fatto che i segnali che vi transitano devono essere in tempo reale, è praticamente impossibile utilizzarla in contemporanea su più apparecchiature. L'interfaccia utilizza gli usuali pin SDI (MISO), SDO (MOSI) e SCK (clock) per i dati, mentre il pin di selezione viene utilizzato in modo differente per selezionare il canale audio tra i due trasmessi; è inoltre presente una linea addizionale di clock, utilizzata come riferimento.

Quindi abbiamo, nominando i pin dal lato master, ovvero dal punto di vista del controller:

- MISO o SDI --> ADCDAT, ovvero i dati in ingresso dal convertitore A/D al controller;
- MOSI o SDO --> DACDAT, ovvero i dati in uscita dal convertitore D/A verso la PIC;
- SCK --> BLCK, ovvero il clock per i singoli bit trasmessi;
- SS --> LRCK, ovvero il clock per differenziare

canale sinistro e destro;

- MCLK --> clock di riferimento, abitualmente 4 volte il SCK.

Anche qui abbiamo dovuto "multiplexare" gli I/O del codec con altrettanti I/O disponibili sulla porta esterna in fondo al controller, e precisamente gli I/O dal 45 al 49.

Utilizzando l'audio questi I/O non saranno quindi disponibili per le proprie applicazioni; per contro, non utilizzando il codec avremo quegli I/O liberi e, soprattutto, potranno essere utilizzati come un'ulteriore interfaccia SPI (chiamata SPI0 da software) totalmente indipendentemente dalla SPI standard. I segnali che transitano da/verso il codec sono visibili nella Fig. 1.

In quest'immagine manca il clock di riferimento (MCLK) che è comunque un clock fisso di frequenza quadrupla rispetto al BLCK. Lasciamo per il momento la descrizione del codec,

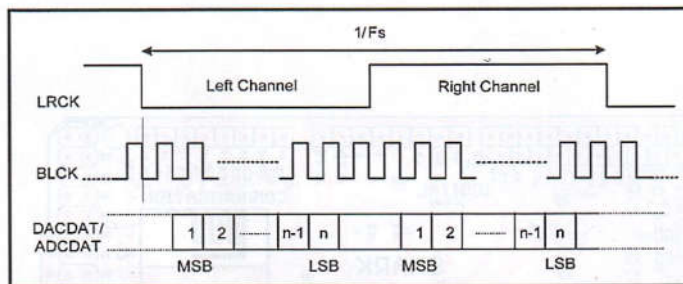


Fig. 1 - Temporizzazioni dei segnali del codec.

che riprenderemo quando tratteremo le librerie software.

GLI HEADER

In questa sezione sono raggruppati tutti i connettori che portano i segnali verso l'esterno della scheda, quindi gli header laterali a standard Arduino, quello doppio in fondo alla scheda, il connettore ISP (che su Fishino Shark non è utilizzato per programmare la scheda, ma inserito per compatibilità con gli shields esistenti), il connettore ICSP (In Circuit Serial Programming) che serve per programmare la scheda senza passare per il bootloader (ad esempio per programmare il bootloader stesso), e sul quale sono disponibili due I/O digitali aggiuntivi, ed il connettore USB.

Come accennato elencando le caratteristiche della scheda, il PIC impiegato nella Shark ha molti, ma non tutti, i pin di I/O 5V tolerant, ovvero in grado di accettare 5V in ingresso.

Dove è stato possibile, abbiamo sfruttato quei pin portandoli all'esterno come I/O digitali; purtroppo in alcuni casi questo non è stato possibile ed abbiamo quindi inserito una protezione in ingresso tramite resistenze e diodi. La protezione consiste in una resistenza da 680 ohm in serie all'ingresso e in un diodo in grado di scaricare l'eccedenza di tensione sulla linea a 3,3 volt, limitando quindi la tensione in ingresso a tale valore.

Sui pin analogici, che non sono 5V tolerant (proprio a causa della circuiteria analogica interna al controller) abbiamo inserito solamente delle resistenze in serie per limitare un eventuale sovracorrente nel caso vi si inseriscano per errore dei valori superiori ai 3,3 volt.

Attenzione, questo non significa che è possibile inserire 5 volt su quel connettore! Le resistenze offrono un minimo grado di protezione, che dovrebbe proteggere il controller per brevi periodi se per errore vi si inseriscono i 5 volt, ma non bisogna farne conto ed utilizzarli stabilmente con quei valori di tensione!

L'integrato dispone infatti di diodi di protezione al suo interno, che fanno le veci di quelli da noi in-

seriti esternamente nei pin digitali protetti; questi diodi sono però in grado di assorbire una corrente molto limitata e per brevi periodi di tempo, quindi non ci si deve fare assolutamente conto.

Nella sezione header troviamo anche un connettore denominato REFSEL, che serve per selezionare la tensione che verrà connessa al pin VREF; è infatti possibile farvi giungere i 5V, nel caso si utilizzino shields "normali" funzionanti a quella tensione, oppure i 3,3V destinati a shield a bassa tensione come quelli utilizzati nell'Arduino DUE.

REALIZZAZIONE PRATICA

Fishino Shark è una scheda realizzata completamente con componenti SMD, alcuni di dimensioni particolarmente ridotte e popolata su entrambi i lati.

Pur utilizzando un PCB a due facce, quindi teoricamente realizzabile con metodi artigianali, ha molte piste sottili e moltissime vie (connessioni tra le due facce) che ne rendono la realizzazione.

Invece il montaggio non è particolarmente complesso, salvo per il codec audio ALC5631Q, che, essendo in package QFN, ha i piedini sotto il corpo, anche se lungo il bordo ed è praticamente impossibile da saldare con un comune saldatore a stilo, ma richiede la stazione ad aria calda.

Suggeriamo quindi ai meno esperti di acquistare la scheda già pronta all'uso, perché è più conveniente. Sul nostro sito mettiamo comunque a disposizione tutti i file necessari a realizzare il PCB.

CONCLUSIONI

Concludiamo qui questa prima puntata dove abbiamo introdotto le peculiarità della nostra scheda Fishino Shark, descrivendone l'hardware ed il funzionamento. Nella prossima puntata vi spiegheremo come gestirla dall'IDE di Arduino e vi proporremo un primo sketch di prova per testarne le potenzialità.



per il MATERIALE

È possibile acquistare la board Fishino Shark (cod. SHARK) presso Futura Elettronica al prezzo di Euro 67,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - Fax: 0331-792287 - www.futurashop.it

PROMOSSO E ORGANIZZATO DA



Camera di Commercio
Roma

CON IL PATROCINIO DI



ROMA



CON IL SUPPORTO DI



IN COLLABORAZIONE CON



REGIONE
LAZIO

ROME

Maker Faire®

THE EUROPEAN EDITION

IL PIÙ GRANDE EVENTO EUROPEO
SULL'INNOVAZIONE

12 - 14
OTTOBRE
FIERA DI ROMA

MAKERFAIREROME.EU
#MFR18

MAIN PARTNER



GOLD PARTNER



MEDIA PARTNER



TECHNICAL
PARTNER



OFFICIAL RADIO



OFFICIAL CARRIER



LOCATION



THANKS TO



COMPAGNI

D'AVVENTURA!

PORTATA
FINO A
3 km
IN SPAZI APERTI



Prezzi IVA inclusa.

COPPIA DI RICETRASMETTORI PMR 446 A 8 CANALI

€ 19,90
Cod. FR698

Comunica a costo zero con i nuovi ricetrasmettitori PMR 446. Ideali per il tempo libero, le escursioni, il lavoro, la caccia, il softair, o come interfono all'interno dell'abitazione.

Coppia di ricetrasmettitori PMR (personal mobile radio) a 8 canali operanti sulla banda 446 MHz, con potenza d'uscita di 0,5 W. Dispongono di 38 toni CTCSS in TX e RX, 10 toni di chiamata, VOX per comunicazione a mani libere, roger beep, display con retroilluminazione, torcia a LED, indicazione batteria scarica, presa da 2,5 mm per cuffia, funzione SCAN e blocco tastiera. Alimentazione tramite 4 batterie da 1,5 volt tipo AAA (non incluse), dimensione 145x55x25 mm, peso 79 grammi. Per utilizzare i ricetrasmettitori è sufficiente la comunicazione di inizio attività ed il pagamento della tassa di concessione annuale.

Scopri la gamma completa su www.futurashop.it

€ 69,00
Cod. G7PRO

- ✓ Potenza di trasmissione: 500 mW (PMR446)
- ✓ 10 mW (LPD)
- ✓ 8 canali (PMR446)
- ✓ 69 canali (LPD)



€ 85,00
Cod. G9

- ✓ Potenza di trasmissione: 500 mW (PMR446)
- ✓ 10 mW (LPD)
- ✓ 24 canali (PMR446)
- ✓ 69 canali (LPD)



MIDLAND

€ 159,00
Cod. CT890

- ✓ Potenza di trasmissione: 5W VHF/4W UHF
- ✓ 999 canali memorizzabili



€ 107,00
Cod. CT690V

- ✓ Potenza di trasmissione: 5W VHF/UHF
- ✓ 128 canali memorizzabili



RICETRASMETTORI VHF-UHF

www.futurashop.it

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

Caratteristiche tecniche di questo prodotto e acquisti on-line su www.futurashop.it

FUTURA ELETRONICA®

GSM SHIELD UNIVERSALE

di MATTEO DESTRO

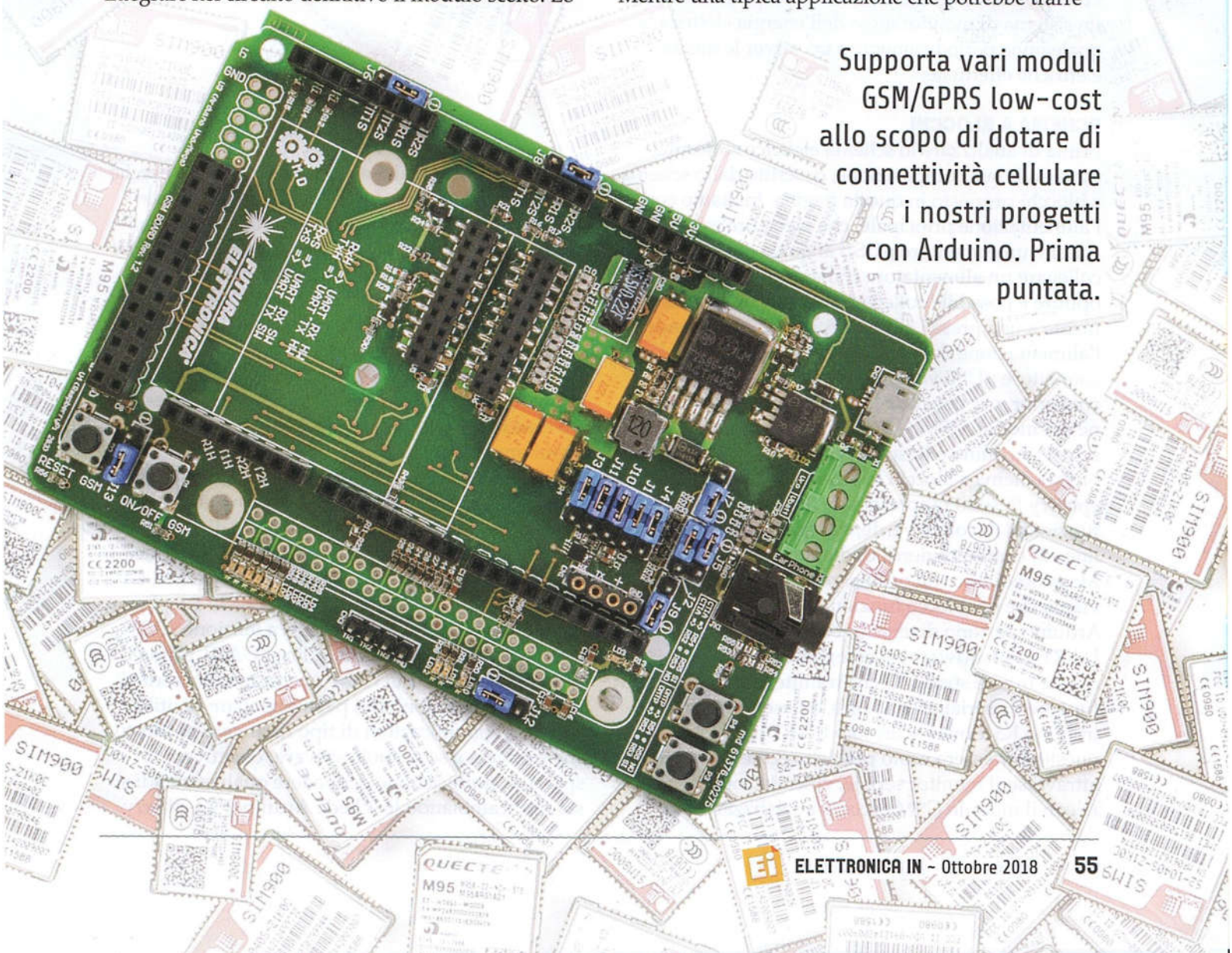
Data la richiesta crescente di apparecchiature elettroniche con connessione alla rete GSM/GPRS, abbiamo deciso di sviluppare un nuovo shield GSM per piattaforma Arduino e Raspberry Pi 3, capace di supportare svariati moduli su breakout-board GSM sviluppate da Futura Elettronica, le quali utilizzano i motori GSM di SimCom, Quectel e Fibocom sulla medesima pin out. Quindi sarà possibile provare e trovare il modulo GSM più adatto alla propria applicazione, sviluppare il firmware sullo shield e poi integrare nel circuito definitivo il modulo scelto. Lo

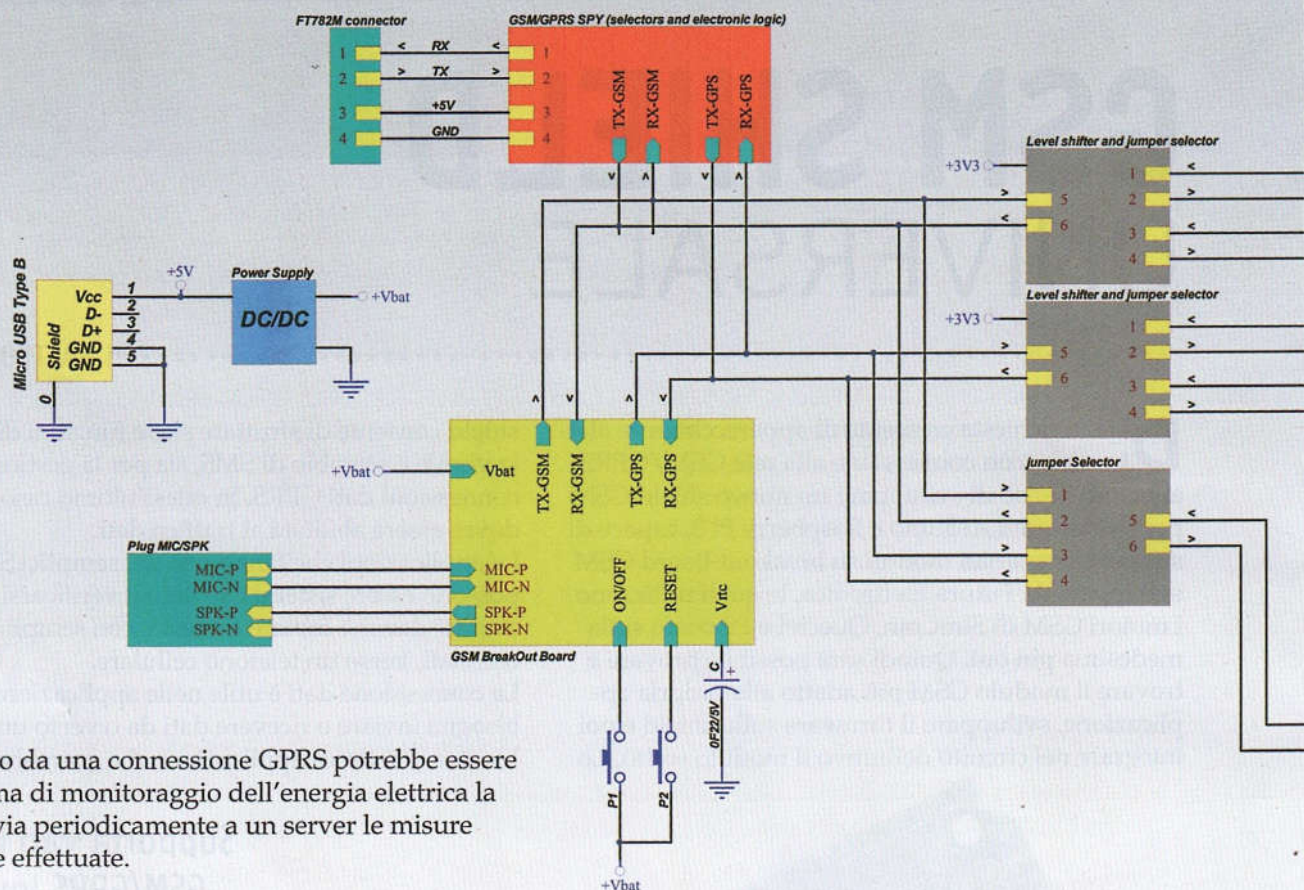
shield consente di sfruttare sia le funzioni di chiamata vocale e scambio di SMS, sia per la gestione delle connessioni dati GPRS. In quest'ultimo caso la SIM dovrà essere abilitata al traffico dati.

Le applicazioni che fanno uso dei semplici SMS possono essere sistemi che dato il verificarsi di un evento devono inviare un testo, con semplici dati correlati, verso un telefono cellulare.

La connessione dati è utile nelle applicazioni in cui bisogna inviare o ricevere dati da o verso un server. Mentre una tipica applicazione che potrebbe trarre

Supporta vari moduli
GSM/GPRS low-cost
allo scopo di dotare di
connettività cellulare
i nostri progetti
con Arduino. Prima
puntata.





vantaggio da una connessione GPRS potrebbe essere un sistema di monitoraggio dell'energia elettrica la quale invia periodicamente a un server le misure elettriche effettuate.

SCHEMA A BLOCCHI

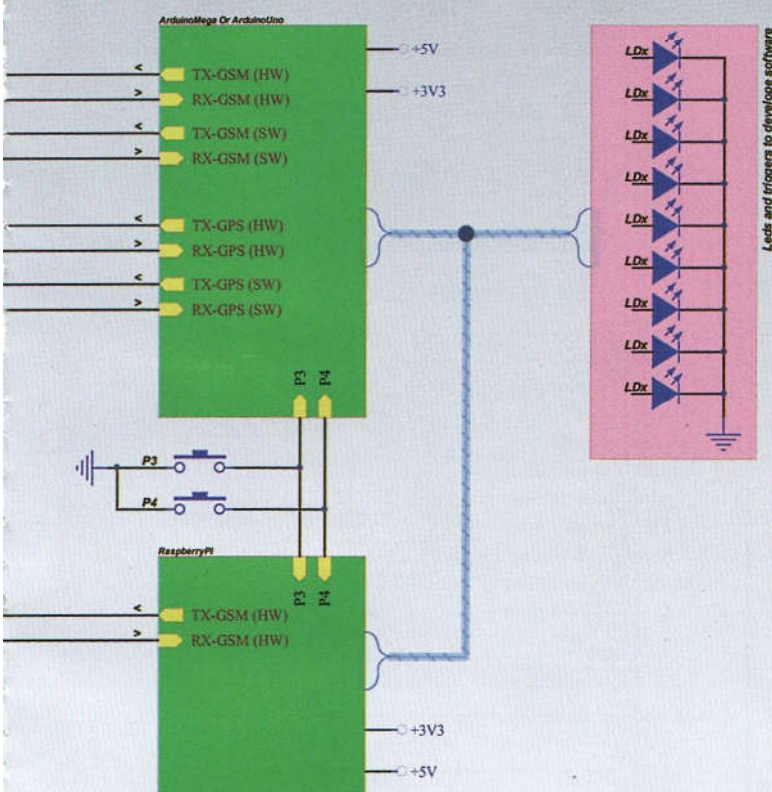
Prima di analizzare lo schema elettrico soffermiamoci sulla sua composizione, descritta dallo schema a blocchi proposto in questa pagina. Iniziamo con l'alimentazione principale, che viene prelevata tramite un connettore μ USB al quale è possibile collegare un alimentatore come quello dei cellulari, oppure è possibile collegare il cavo USB direttamente a una presa USB tipo A del PC, prendendo l'alimentazione dallo stesso. I 5 volt presenti sul connettore μ USB vengono portati sia al DC/DC converter, per generare la tensione di alimentazione dei moduli GSM (+VBAT) sia agli header per le schede Arduino e Raspberry Pi.

Nello schema a blocchi le schede Arduino e Raspberry Pi sono rappresentate da due blocchi funzionali di colore verde, dai quali partono alcune linee di comunicazione, di diagnostica e debug. Ma procediamo con ordine: per prima cosa osserviamo che dal blocco funzionale corrispondente ad Arduino partono diverse linee di comunicazione UART, sia hardware sia assegnate da software; queste ultime sono I/O che tramite librerie possono fungere da seriali e sono utili, ad esempio su Arduino Uno R3, dove c'è un solo UART hardware. Se questo dev'essere usato per debuggare il firmware attraverso il monitor seriale, volendo comunicare con il modulo GSM occorre inizializzare una

seriale virtuale. Qualcuno si starà chiedendo perché abbiamo predisposto due UART? La scelta nasce dal fatto che ci possono essere moduli GSM dotati di due UART: ad esempio il SimCom SIM928A, che integra un ricevitore GPS e dispone di una seriale per ricevere i comandi AT per la gestione della sezione GSM/GPRS e l'altra per la lettura dei dati GPS NMEA sul posizionamento.

Per quanto riguarda Raspberry Pi, questa mette a disposizione un solo UART hardware; in questo caso non è possibile avere due comunicazioni simultanee.

Le linee UART, nel caso delle schede Arduino, prima di giungere ai moduli GSM passano attraverso dei traslatori di livello che permettono di accoppiare linee dati con tensioni di lavoro a +5V con linee dati con tensioni di lavoro a +3,3V. Questo non è necessario per quanto riguarda le schede Raspberry Pi, in quanto queste lavorano già a tensione +3,3V. Oltre ai traslatori di livello, le linee UART prima di giungere ai moduli GSM incontrano una serie di jumper a 3 posizioni utili per selezionare se utilizzare una comunicazioni di tipo hardware piuttosto che software e se si vuole attivare/disattivare la spia di comunicazione per vedere quali comandi si stanno scambiando le schede Arduino/Raspberry



Pi con i motori GSM. È anche possibile instradare i segnali in modo da inviare i comandi AT direttamente da PC, questo è molto utile durante la fase di studio dei comandi per verificarne il corretto funzionamento prima di implementarli in codice nelle schede Arduino e Raspberry Pi. Queste sezioni di level shifting e instradamento dei segnali sono rappresentate da dei blocchi di colore grigio (sono in tutto tre blocchi distinti, due nominati "Level shifter and jumper selector" mentre il terzo è solo un "Jumper selector").

La sezione dei motori GSM è rappresentata dal blocco di colore giallo "GSM BreakOut Board" al quale giungono le linee di comunicazione discusse in precedenza. A questo blocco, oltre alle linee dati, giunge anche l'alimentazione +VBAT generata dal DC/DC converter e l'alimentazione Vrtc resa disponibile dal condensatore a foglie d'oro da 0,22F usato per tenere alimentato il Real Time Clock Calendar presente sui moduli GSM in caso di mancata alimentazione primaria. La carica del condensatore viene garantita dalla stessa elettronica dei moduli GSM, nel momento in cui questa viene a mancare il condensatore provvede a tenere attivo il RTCC fino a sua completa scarica.

Ai moduli GSM vengono collegati anche due

pulsanti i quali servono per accendere/spegnere il motore GSM, pulsante P1, oppure per resettare il motore GSM, pulsante P2. Questi pulsanti vengono utili quando si deve agire manualmente sui motori, ad esempio accenderli per poi inviare dei comandi AT da PC. La durata dell'impulso di accensione/spegnimento, piuttosto che la durata dell'impulso di reset dipendono dal motore stesso. Quindi è necessario consultare i datasheet per i dettagli, ovviamente si parla sempre di impulsi di durata minima. Concludiamo la carrellata con le linee MIC e SPK, linee differenziali, le quali servono per collegare rispettivamente un microfono e uno speaker utili durante le chiamate foniche.

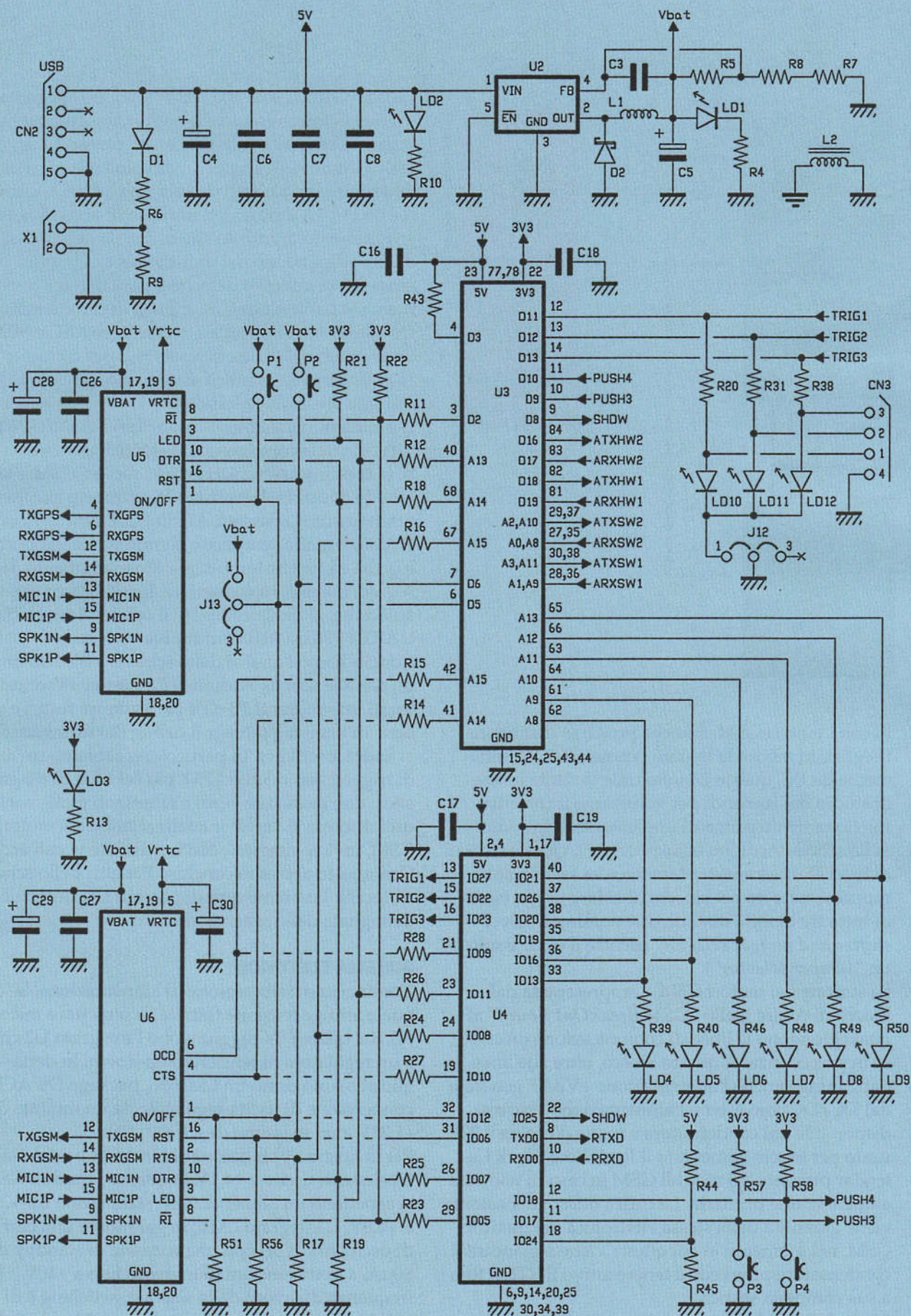
Esaminiamo ora il blocco arancione "GSM/GPRS SPY" il quale come accennato prima serve per spiare la comunicazione seriale tra i motori GSM e le schede Arduino/Raspberry Pi, piuttosto che inviare direttamente i comandi AT da PC. Il blocco funzionale è molto semplice e comprende delle porte logiche e qualche jumper per l'instradamento dei segnali nonché un connettore 4 poli, blocco azzurro adiacente, al quale collegare il convertitore USB/UART FT782M della Futura Elettronica.

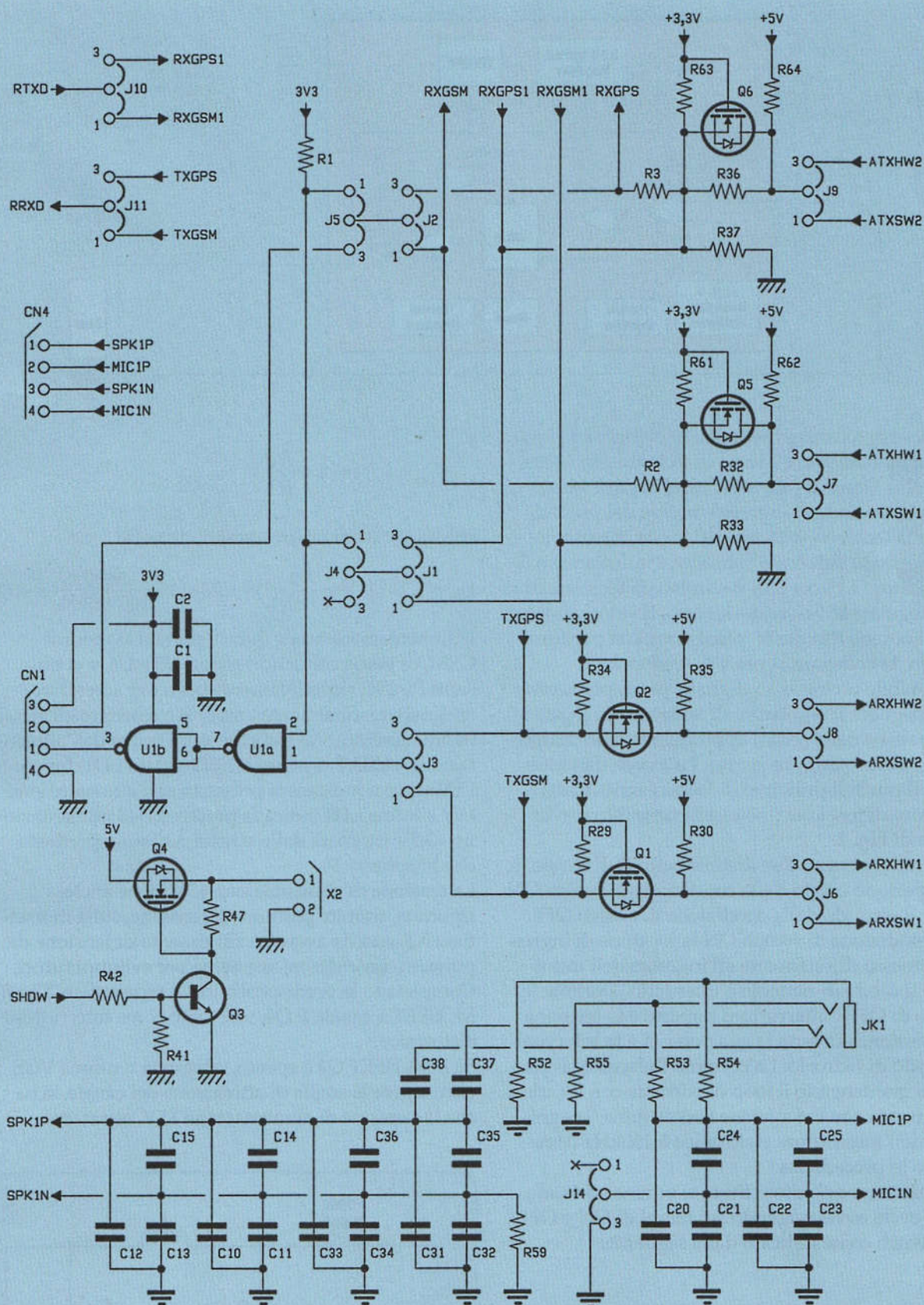
Concludiamo l'analisi dello schema a blocchi dicendo che alle schede Arduino/Raspberry Pi sono collegati due pulsanti P3 e P4 per sviluppi futuri e una serie di linee digitali per il debug del firmware che si andrà a scrivere. In particolare, abbiamo tre linee di trigger, con relativi LED, più sei LED di diagnostica: due rossi, due verdi e altrettanti gialli; sono utili durante le fasi di inizializzazione dei motori GSM, invio e ricezione SMS, chiamate vocali ecc. Con questo abbiamo concluso l'analisi dello schema a blocchi. Possiamo quindi passare alla descrizione dettagliata dello schema elettrico.

SCHEMA ELETTRICO

Cominciamo dalla sezione di alimentazione, la quale attinge tensione tramite il connettore microUSB, dal quale i 5V raggiungono l'integrato U2 che è un regolatore di tensione step-down. In dettaglio abbiamo usato un LM2596, package D²PACK, con tensione di uscita regolabile da un minimo di +1,23V a un massimo di +37V.

Per la nostra applicazione abbiamo impostato la tensione di uscita a +4,1V in quanto i moduli GSM si aspettano un'alimentazione stabilizzata tra +3,2V a +4,8V. L'integrato LM2596 garantisce una corrente di uscita fino a 3A con una corrente di standby di 80μA, accetta tensioni di ingresso fino a +40V e la frequenza di lavoro della sezione switching è di 150kHz. La Fig. 1 mostra lo schema a blocchi del





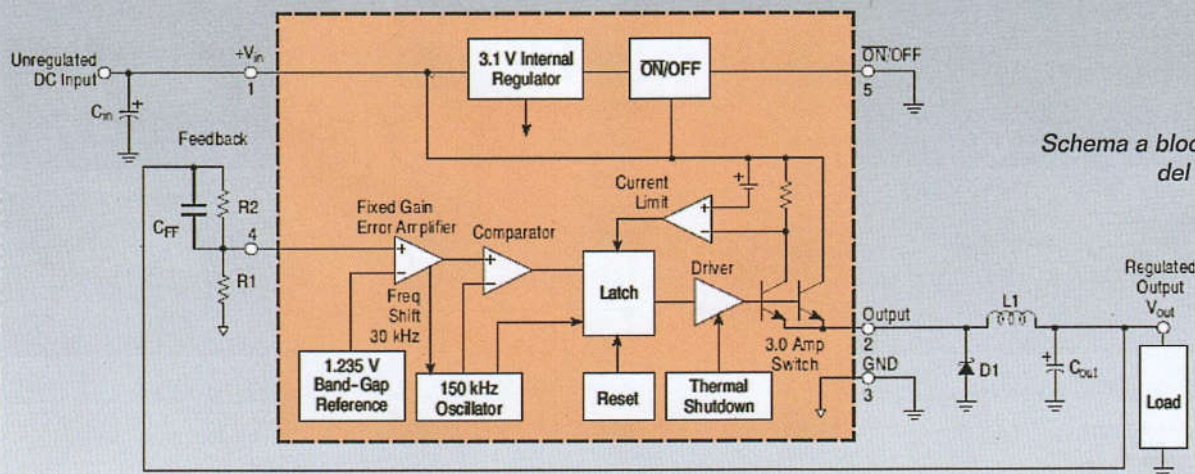


Fig. 1
Schema a blocchi interno
del regolatore.

regolatore di tensione mettendo in evidenza la logica di funzionamento e i vari pin di ingresso/uscita e controllo. Come si può osservare, al pin 1 viene portata la tensione di ingresso mentre dal pin 2 si preleva la tensione switching di uscita che tramite la rete formata dal diodo Schottky, l'induttanza e il condensatore a bassa ESR restituisce la tensione di uscita regolata al valore desiderato. Il valore di tensione si ottiene tramite la retroazione con partitore resistivo da collegare al pin 4 di feedback.

È disponibile un ingresso digitale, pin 5, per accendere/spegnere il regolatore di tensione. La logica è attiva bassa e nella nostra applicazione lo forziamo a massa ovvero sempre acceso. La massa dell'integrato è il pin 3. Il principio di funzionamento del regolatore di tensione può essere riassunto con lo schema di Fig. 2.

L'integrato opera su due distinti periodi di tempo, il primo periodo è dato dalla condizione di switch ON mentre il secondo dalla condizione di switch OFF. Nella condizione di switch ON la tensione di ingresso è connessa direttamente all'ingresso dell'induttanza e il diodo di ricircolo è interdetto. Durante il periodo di OFF l'interruttore è aperto e la tensione sull'induttanza inverte la sua polarità e fa intervenire il diodo di ricircolo. La corrente fluisce attraverso il diodo mantenendo il loop di corrente con il carico. La frequenza con cui avviene l'accensione/spegnimento dell'interruttore elettronico è 150kHz come indicato in precedenza.

La regolazione del convertitore si ottiene variando il duty-cycle ovvero agendo sui tempi di ON e OFF dello switch come indicato dalla seguente:

$$d = \frac{t_{on}}{T}$$

Fatta questa premessa, calcoliamo i valori che devono assumere le resistenze di feedback per impostare la tensione di uscita al regolatore. La formula da seguire è la seguente (con riferimento alla Fig. 1):

$$V_{out} = 1,23 * \left(1 + \frac{R2}{R1}\right)$$

che applicato al nostro circuito diventa:

$$V_{out} = 1,23 * \left(1 + \frac{R5}{R7 + R8}\right) = 1,23 * \left(1 + \frac{5600}{200 + 2200}\right) = 4,1 V$$

L'alimentazione viene quindi portata ai moduli GSM, ovvero i connettori siglati U5 e U6, e ai pulsanti P1 e P2 rispettivamente usati per accendere/spegnere manualmente i moduli e resettare gli stessi (manualmente). Ad indicare la presenza dell'alimentazione +VBAT ci pensa il LED siglato LD1. Invece LD2 serve a indicare la presenza dell'alimentazione +5V e infine LD3 indica la presenza dell'alimentazione +3,3V ritornata dalle schede Arduino, piuttosto che Raspberry Pi.

La tensione di alimentazione +5V viene anche riportata, tramite partitore di tensione, sulla morsettieria X1, così da avere un riferimento di tensione da portare a un'elettronica esterna per sviluppi futuri. Completano la predisposizione la morsettieria X2 e il MOSFET a canale P Q4, utilizzato come interruttore elettronico.

Se il MOSFET Q4 è spento, ovvero la tensione VGS non supera la soglia di attivazione del canale, si ha che la tensione di alimentazione +5V, attraverso

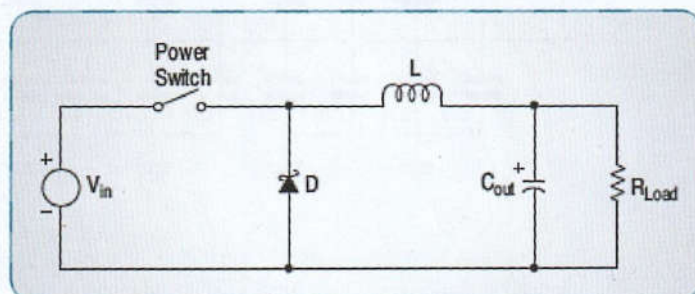


Fig. 2 - Schema di principio del regolatore.

il diodo interno del MOSFET, raggiunge il pin 1 della morsettiera X2. Ora se si porta alto il segnale "shutdw" succede che Q3 si accende portando a massa il gate di Q4 e, grazie alla tensione presente sul pin 1 di X2, la VGS supera la soglia attivando il canale di Q4. In questa condizione nel momento in cui viene a mancare l'alimentazione principale (+5V) e se sulla morsettiera è presente una tensione +5V generata da una scheda ausiliaria si avrà che la nostra elettronica continuerà a funzionare fintantoché non verrà portata bassa la linea "shutdw".

Passiamo ora a descrivere le interconnessioni presenti sia sulle schede Arduino, sia sulle schede Raspberry Pi. Le schede Arduino supportate sono Arduino Uno R3, Arduino Mega 2560 R3 e le schede Fishino (Uno e Mega). Nel caso in cui si usino le schede Fishino (usare schede con implementato ultimo firmware che libera I/O 7) è necessario ricordare che gli I/O 10, 11, 12 e 13 sono usati per il WiFi e nel caso si usi la scheda microSD anche I/O 4 deve essere libero. Dato che agli I/O 10, 11, 12 e 13 avevamo previsto di collegare il pulsante P4 e tre trigger per il debug si dovrà tenere conto di ciò quando si andrà ad usare la scheda Fishino Uno. Se invece si usa la scheda Fishino Mega gli I/O da lasciare liberi sono il 4, 10, 50, 51 e 52 (anche in questo caso usare schede con implementato ultimo firmware che libera I/O 7). Se si devono liberare gli I/O 11, 12 e 13 spostare il jumper J12 sulla posizione 2-3.

Per quanto riguarda le schede Raspberry Pi sono supportate: Raspberry Pi 2 model B+, Raspberry Pi 3 model B e Raspberry Pi 3 model B+.

Cominciamo dalle schede Arduino, per comodità abbiamo usato come riferimento di partenza la pin-out della scheda Arduino Mega con evidenziati in azzurro i pin corrispondenti della scheda Arduino Uno. Se si usa la scheda Arduino Mega è possibile scegliere se sfruttare le due UART hardware, UART1 e UART2, mappate sui pin di I/O 16, 17, 18 e 19 oppure le due UART software che sono mappate sui pin di I/O A8 (RX UART2), A9 (RX UART1), A10 (TX UART2) e A11 (TX UART1). I corrispondenti I/O per la scheda ArduinoUno sono A0, A1 A2 e A3. Le UART di tipo software, per quanto riguarda il segnale di ricezione, devono essere mappate obbligatoriamente sui pin di I/O che supportano il port change interrupt. Quindi prestate attenzione nello sviluppare eventuale hardware per le vostre applicazioni.

I segnali TX e RX delle due UART, sia hardware che software, devono subire un adattamento di livello in quanto le schede Arduino lavorano con livelli logici a +5V mentre i moduli GSM lavorano con livello

a +3,3V. Quindi il segnale TX, prima di giungere al motore GSM, viene adattato tramite l'utilizzo di un mosfet a canale N opportunamente collegato. Stesso discorso per il segnale RX che dal modulo GSM giunge alla scheda Arduino. Per capire meglio quanto detto esaminiamo la connessione delle linee TX e RX, sia hardware che software, usate per inviare i comandi AT per la gestione della sezione GSM. Come si può osservare dallo schema elettrico, lato Arduino, le linee interessate sono identificate come segue "ARD_RXD1_HW", "ARD_TXD1_HW", "ARD_RXD1_SW" e "ARD_TXD1_SW".

Queste giungono a dei jumper di selezione, J6 e J7, necessari per decidere se sfruttare una connessione seriale di tipo hardware oppure software. Quindi se i jumper sono in posizione 1-2 viene identificata una connessione UART software mentre se i jumper sono in posizione 2-3 viene identificata una connessione UART hardware.

Per quanto riguarda il segnale TX Arduino, che sia hardware o software, giunge al MOSFET Q5, il quale adatta il livello di tensione a +3,3V compatibile con l'elettronica dei moduli GSM. Oltre al MOSFET Q5 sono state predisposte anche due resistenze, non montate, in configurazione partitore di tensione per adattare il livello di tensione da +5V a +3,3V (Solo per la linea TX). Per quanto riguarda il segnale RX, sempre lato Arduino, abbiamo che questo viene portato al drain del MOSFET Q1 che così collegato permette di adattare il segnale proveniente dal motore GSM a +3,3V al livello di tensione +5V di Arduino. Questo tipo di adattamento lo abbiamo già usato in tante altre applicazioni.

Lo stesso ragionamento è stato applicato alle linee TX e RX per la gestione dei comandi AT del GPS e quindi facenti capo ai jumper J8 e J9.

Come accennato, è possibile spiare i comandi AT che vengono inviati al modulo GSM/GPS e le relative risposte sfruttando l'elettronica formata dall'integrato U1 e dai jumper J1, J2, J3, J4 e J5.

A completare la spia il solito convertitore USB/Seriale FT782M della Futura Elettronica configurato per lavorare con I/O a livello logico +3,3V e inserito nell'apposito connettore CN1. Grazie a questa elettronica e a una opportuna configurazione dei jumper è possibile fare le seguenti:

- 1) spiare i comandi AT inviati al modulo GSM/GPS da scheda Arduino/Raspberry Pi;
 - 2) spiare le risposte dei comandi AT inviati alla scheda Arduino/Raspberry Pi;
 - 3) inviare i comandi AT direttamente da PC escludendo le linee TX e RX di Arduino/Raspberry Pi.
- Quindi se si vogliono spiare i comandi AT, e le rela-

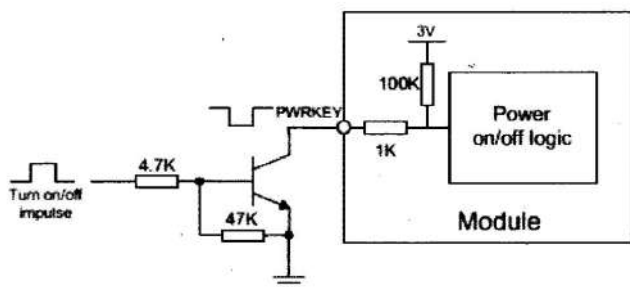


Fig. 3 - Comando della linea di accensione del modulo GSM.

tive risposte, inviati al modulo GSM i jumper J1, J2, J3, J4 e J5 devono essere in posizione 1-2. Se invece si vogliono spiare i dati ricevuti dal modulo GPS i jumper J1, J2 e J3 devono essere in posizione 2-3 mentre i jumper J4 e J5 devono essere in posizione 1-2. Se invece si volesse inviare i comandi AT direttamente da PC, ad esempio per studiare nuovi comandi da implementare negli sketch Arduino o Raspberry Pi, si devono configurare i jumper nel seguente modo: jumper J2 e J3 in posizione 1-2, jumper J4 e J5 in posizione 2-3 e jumper J1 indifferente (*configurazione per invio dei comandi AT a sezione GSM*). Oppure jumper J2, J3, J4 e J5 in posizione 2-3 e J1 indifferente (*configurazione per lettura dati GPS*).

Per escludere la possibilità di inviare comandi AT da PC in modo permanente è sufficiente non montare le due resistenze da 0 ohm R2 e R3. In questo modo è impossibile inviare comandi AT da PC verso il motore GSM/GPS in quanto la linea risulta interrotta. Consigliamo comunque di montare sempre la sezione spia, dato che risulta utile durante le fasi di studio e apprendimento dei comandi AT.

Inoltre data la possibilità di sfruttare un terminale PC per l'invio/ricezione dei comandi AT si ha un metodo di studio veloce e efficace che vi svincola dagli eventuali bug introdotti durante la scrittura di codice C per lo sviluppo o integrazione di una libreria Arduino/Raspberry Pi. In altre parole, prima si studia il comando AT che si vuole implementare sfruttando un terminale da PC e poi, solo dopo averne capito il funzionamento, si implementa nella propria libreria.

Tornando alle linee di comunicazione TX e RX, quelle che giungono dalla Raspberry Pi hanno già livelli compatibili con i moduli GSM e quindi non necessitano di adattamento di linea. Tuttavia si deve decidere tramite i jumper J10 e J11 se pilotare la sezione GSM piuttosto che quella GPS. Se J10 e J11 in posizione 1-2 si comunicherà con la sezione GSM, viceversa se in posizione 2-3 si comunicherà con sezione GPS.

Oltre alle linee TX e RX abbiamo predisposto anche

le linee di controllo le quali però, attualmente, non sono utilizzate nella nostra libreria, ovvero le linee "CTS", "DCD", "RTS" e "DTR".

Molto più importanti le linee "ON/OFF" e "RST" che rispettivamente servono per accendere/spegnere il modulo GSM/GPS oppure resettare gli stessi. La Fig. 3 evidenzia come deve essere predisposta l'elettronica necessaria per accendere/spegnere il modulo GSM, come si può vedere l'impulso di accensione dovrà essere a logica positiva e la sua durata dovrà essere superiore a un secondo.

La Fig. 4 mostra il diagramma temporale di accensione del modulo GSM e più precisamente mostra l'andamento che deve avere il segnale PWRKEY che si trova sul collettore del transistor. Quindi affinché l'operazione di accensione del GSM vada a buon fine, è necessario che, dopo avere dato alimentazione, si attenda almeno 500ms prima di inviare l'impulso di accensione (che deve durare più di 1 secondo).

Oltre all'accensione del GSM sfruttando l'apposita linea, abbiamo predisposto il pulsante P1 che assolve al medesimo compito; fornendo il livello logico 1 perché deve attivare il transistor che internamente al modulo pone a zero il PWRKEY.

La Fig. 5 mostra l'andamento che deve avere il segnale PWRKEY allo spegnimento del modulo GSM: l'impulso a zero logico deve avere una durata superiore al secondo ma inferiore a 33 secondi. I diagrammi temporali appena mostrati facevano riferimento a un motore SIM800C. Se si usano altri motori, ad esempio un M95 della Quectel, la durata degli impulsi di accensione e spegnimento potrebbe essere diversa. In questo caso l'impulso di spegnimento deve avere una durata compresa tra 700ms e al massimo un secondo. Quindi è sempre buona norma consultare il datasheet del modulo.

Oltre alla linea di accensione e spegnimento, abbiamo predisposto una linea di reset ("RST") la quale serve appunto a resettare il motore GSM. Il pulsante

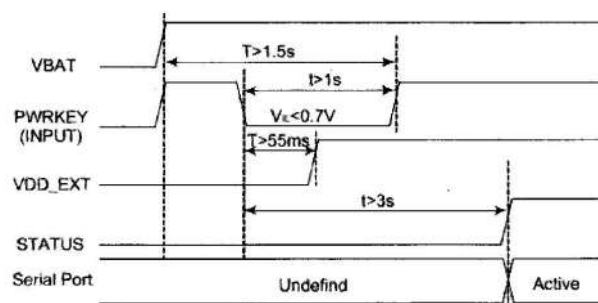


Fig. 4 - Temporizzazioni dei segnali riguardanti il GSM.

P2 assolve alla funzione di reset manuale.

Concludiamo la rassegna con la linea "RI", detta "Ring Indicator", la quale viene portata all'ingresso INT0 (ArduinoUno) o INT4 se fosse ArduinoMega. Questa linea, attiva a livello logico basso, indica la ricezione di un SMS oppure di una chiamata fonica. Se prendiamo come riferimento un motore SIM800C abbiamo che durante la ricezione di un SMS la linea va bassa per 120ms per poi ritornare alta. Invece se si riceve una chiamata fonica la linea va bassa e ritorna alta soltanto se si risponde alla chiamata fonica con apposito comando AT oppure se si rifiuta la chiamata sempre con apposito comando AT.

La Fig 6 descrive l'andamento del segnale sulla linea RI (Ring Indicator) per quanto riguarda la ricezione di un SMS mentre la Fig. 7 mostra l'andamento durante la ricezione di una chiamata vocale.

Notate il jumper J13, che può essere usato per forzare in boot i moduli GSM nel caso in cui sia necessario aggiornargli il firmware.

I moduli GSM solitamente presentano una sezione audio cui è possibile collegare un microfono e degli altoparlanti per poter effettuare delle chiamate vocali; solitamente si usano delle cuffie con microfono dotate di jack audio del microfono separato dal jack degli altoparlanti, oppure (sono le più comuni) con un unico jack, che però non è standardizzato ma esiste in due versioni: CTIA e OMTP (Fig. 8).

La differenza è nel contatto del microfono, che si trova in posizioni differenti. La nostra elettronica è predisposta per funzionare sia con il formato OMTP che con il formato CTIA: montando le resistenze R54 e R55 viene selezionato lo standard CTIA, mentre se si montano le resistenze R52 e R53 viene selezionato lo standard OMTP. Il connettore utilizzato SJ-43514, JK1, è un jack da 3,5mm della CUI-INC il quale ha la pin-out evidenziata dalla Fig. 8, dove notate la corrispondenza esistente tra il jack in formato OMTP/CTIA e i contatti del connettore, come già ampiamente detto l'unica differenza tra i due formati è il fatto che la linea microfonica e la GND sono invertite l'una rispetto all'altra soluzione.

Completano la sezione audio una serie di condensatori da collegare sulle linee differenziali degli altoparlanti e sulle linee differenziali del microfono. Per quanto riguarda i condensatori degli altoparlanti, ci sono due blocchi; uno da collegare vicino al GSM e l'altro vicino al connettore. Il segnale SPK_P viene portato tramite i condensatori C37 e C38 ai due altoparlanti, destro e sinistro, del connettore audio.

Non ci rimane che descrivere le linee di trigger e debug, utili per lo sviluppo del codice sia a livello di libreria sia a livello di sketch. Sulla scheda Arduino

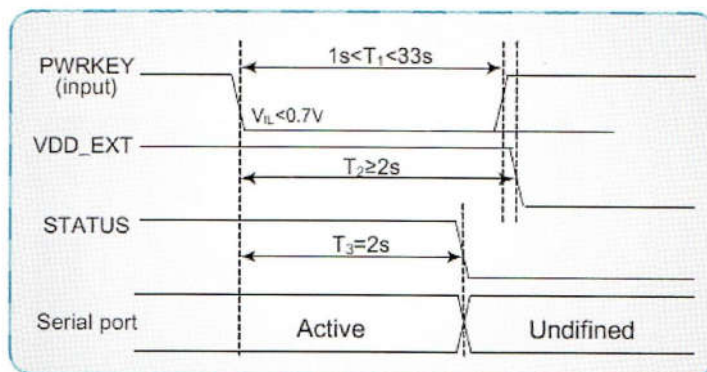


Fig. 5 - Segnale PWRKEY allo spegnimento del GSM.

sono state predisposte le linee di trigger "Trigger1, 2 e 3" e portate tramite resistenza serie al connettore CN3 al quale ci si può connettere con una sonda oscilloscopio a seconda dei casi. Le linee di trigger fanno capo ai pin di I/O 11, 12 e 13. Ricordiamo che nel caso delle schede Fishino tali pin sono condivisi con la sezione WiFi. Completano una serie di LED collegati agli I/O 32, 33, 34, 35, 36 e 37 e disponibili solo sulle schede Arduino Mega e Fishino Mega. Sia le linee di trigger che le linee dei LED sono state portate alla scheda Raspberry Pi.

LAYOUT DEL PCB

Diamo un breve sguardo al layout della scheda, proposto nella Fig. 9 che mostra le sezioni ognuna distinta di un colore:

- sezione arancione, dove trovano posto l'alimenta-

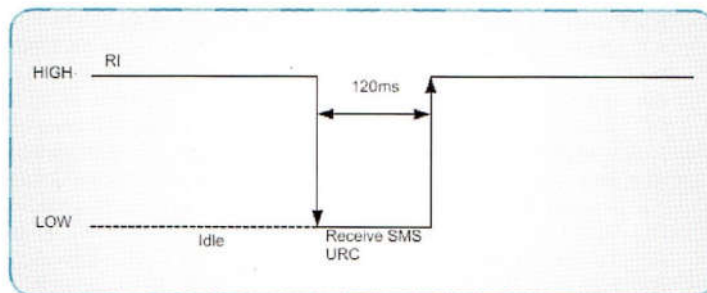


Fig. 6 - Il segnale sulla linea RI del GSM all'arrivo di un SMS.

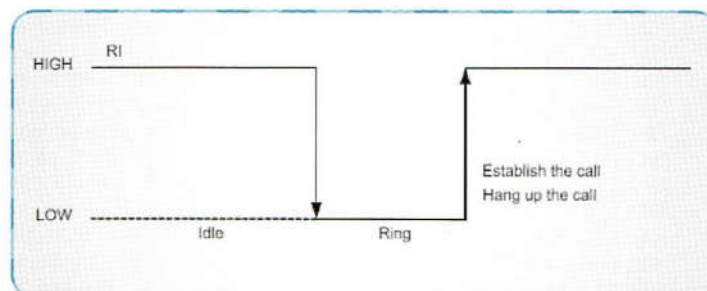
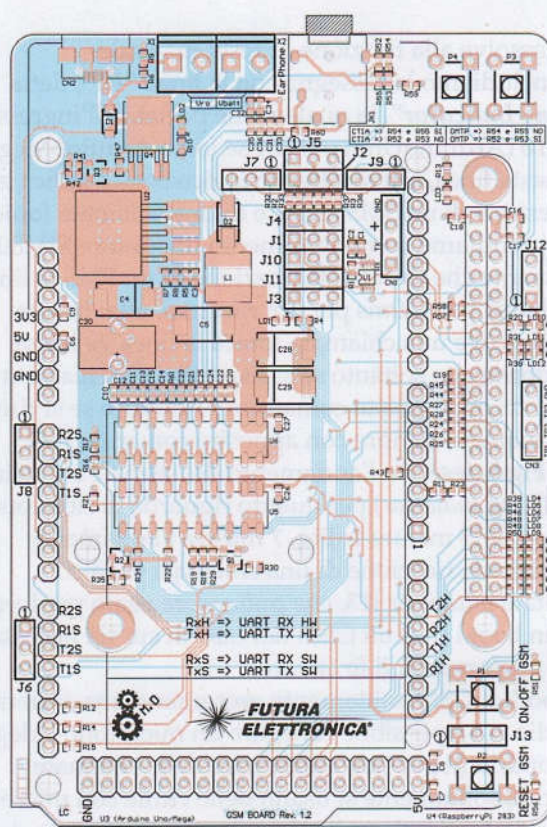
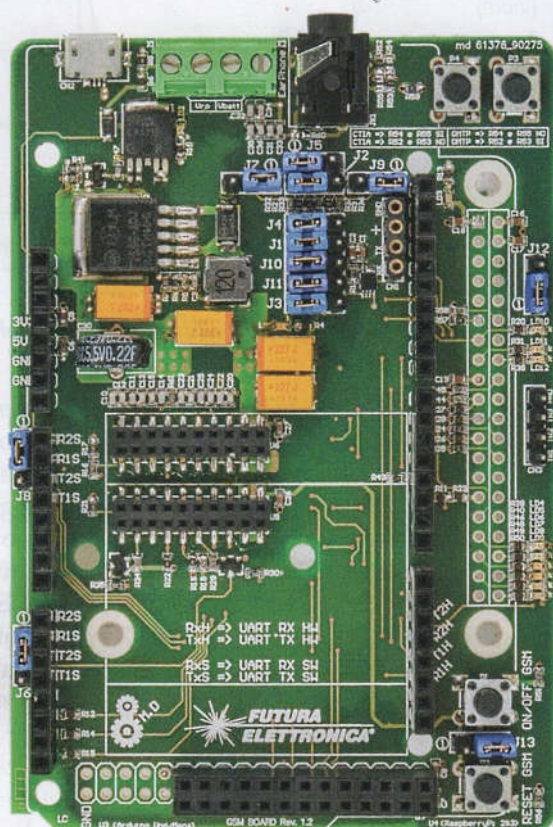


Fig. 7 - Il segnale sulla linea RI del GSM all'arrivo di una telefonata.

[piano di MONTAGGIO]



Elenco Componenti:

R1: 100 kohm 1% (0603)
 R2, R3: 0 ohm (0603)
 R4, R6, R10: 1,2 kohm 1% (0603)
 R5: 5,6 kohm 1% (0603)
 R7: 2,2 kohm 1% (0603)
 R8: 200 ohm 1% (0603)
 R9: 10 kohm 1% (0603)
 R11, R12: 220 ohm 1% (0603)
 R13: 1,2 kohm 1% (0603)
 R14, R15: 220 ohm 1% (0603)
 R16, R18: 5,6 kohm 1% (0603)
 R17, R19: 10 kohm 1% (0603)
 R20: 1,2 kohm 1% (0603)
 R21, R22: 10 kohm 1% (0603)
 R23: 220 ohm 1% (0603)
 R24, R25: 0 ohm (0603)
 R26÷R28: 220 ohm 1% (0603)
 R29, R30: 10 kohm 1% (0603)
 R31: 1,2 kohm 1% (0603)
 R32, R33: -
 R34, R35: 10 kohm 1% (0603)
 R36, R37: -
 R38÷R40: 1,2 kohm 1% (0603)
 R39, R40: 1,2 kohm 1% (0603)
 R41, R45: 10 kohm 1% (0603)
 R42: 36 kohm 1% (0603)
 R43: 220 ohm 1% (0603)
 R44: 5,6 kohm 1% (0603)
 R46, R48÷R50: 1,2 kohm 1% (0603)
 R47: 22 kohm 1% (0603)
 R51: 10 kohm 1% (0603)

R52, R53, R59: -
 R54, R55: 0 ohm (0603)
 R56÷R58: 10 kohm 1% (0603)
 R60: 0 ohm (0603)
 R61÷R64: 10 kohm 1% (0603)
 C1: 100 nF 16 VL ceramico (0603)
 C2: 10 µF 16 VL ceramico (0603)
 C3: 10 nF 16 VL ceramico (0603)
 C4, C5: 220 µF 16 VL tantalio (E)
 C6, C7: 100 nF 16 VL ceramico (0603)
 C8, C9: 10 µF 16 VL ceramico (0603)
 C10, C11: 33 pF 16 VL ceramico (0603)
 C12, C13, C15: 10 pF 16 VL ceramico (0603)
 C14: 33 pF 16 VL ceramico (0603)
 C16÷C19: 10 µF 16 VL ceramico (0603)
 C20, C21: 10 pF 16 VL ceramico (0603)
 C22÷C23: 33 pF 16 VL ceramico (0603)
 C24: 10 pF 16 VL ceramico (0603)
 C25: 33 pF 16 VL ceramico (0603)
 C26, C27: 100 nF 16 VL ceramico (0603)
 C28, C29: 220 µF 16 VL tantalio (E)
 C30: 0,22 F 5 VL elettrolitico
 C31, C32: 33 pF 16 VL ceramico (0603)
 C32: 33 pF 16 VL ceramico (0603)
 C33, C34: 10 pF 16 VL ceramico (0603)
 C35: 33 pF 16 VL ceramico (0603)
 C36: 10 pF 16 VL ceramico (0603)
 C37, C38: 10 µF 16 VL ceramico (0603)
 LD1, LD6, LD7: LED giallo (0603)
 LD2, LD4, LD5: LED rosso (0603)
 LD3, LD8, LD9: LED verde (0603)

LD10÷LD12: LED rosso (0603)
 P1÷P4: Microswitch
 D1: MM5D4148T1G
 D2: B340A-13-F
 Q1, Q2: BSS123
 Q3: BC817
 Q4: SPD50P03LG
 Q5, Q6: BSS123
 U1: SN74LVC2G00
 U2: LM2596DSADJG
 U3: Arduino Mega 2560 rev.3
 U4: Raspberry Pi 3 B+
 U5: Modulo GSM/GPS SIM928
 U6: Modulo GSM G510/M95/SIM900
 L1: VLS6045EX-150M

Varie:

- Connettore Micro-USB B
- Presa Jack 3.5mm da CS
- Strip maschio 3 vie (13 pz.)
- Strip maschio 4 vie (2 pz.)
- Jumper (13 pz.)
- Connettore 2x10 poli passo 2 mm femmina (2 pz.)
- Strip M/F Arduino 8 vie (5 pz.)
- Strip M/F Arduino 10 vie
- Strip M/F Arduino 2x18 vie
- Strip M/F Raspberry Pi 2x20 vie
- Strip femmina 4 vie
- Morsetto 4 vie
- Circuito stampato S1416 (110x158mm)

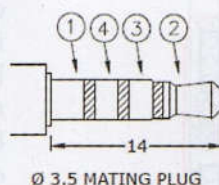
(CTIA)

Apple - recent Android



(OMTP)

All old Android



Model No.	SJ-43514	OMTP	CTIA
Schematic			
PIN			
1	sleeve	MICROPHONE	GND
2	tip	LEFT SPEAKER	LEFT SPEAKER
3	ring 1	RIGHT SPEAKER	RIGHT SPEAKER
4	ring 2	GND	MICROPHONE

Fig. 8 - A sinistra, spinotti audio tipo CTIA e OMTP; qui sopra, piedinatura del jack CUI-INC.

zione portata dal connettore microUSB, la morsetti-
era per collegare un'eventuale sistema di backup
a batteria, il jack audio e due pulsanti ausiliari per
sviluppi futuri;

- **sezione verde**, dove trovano posto i jumper di
selezione delle varie modalità di comunicazione e
spia (trovate una tabella riassuntiva più avanti);
- **sezione grigia**, dove trovano posto i due connet-
tori per l'alloggiamento delle breakout board GSM;
- **sezione rossa**, dove trovano posto i led di trigger
e diagnostica nonché il connettore a pettine per il
collegamento di una eventuale sonda oscilloscopio;
- **sezione blu**, dove trovano posto i due pulsanti
usati per l'accensione/spegnimento del motore
GSM nonché il pulsante di reset.

Lo shield è disponibile in versione già montata, tut-
tavia chi vuole cimentarsi nel montaggio dei com-
ponenti sulla scheda suggeriamo di partire dall'in-
tegrato U11 seguito da Q1 e U2. Poi vanno montati
tutte le resistenze e i condensatori SMD in case 0603,
i due connettori U5 e U6 e successivamente la rima-
nente componentistica SMD a partire dai LED. Per
ultimo montate il jack audio, la morsetti-
era e tutti i jumper e pulsanti. Lasciate per ultimi i connettori di
interconnessione delle schede Arduino e Raspberry
Pi.

La **Tabella 1** riassume i jumper e come devono esse-
re impostati.

COMANDI AT DA PC TRAMITE SPIA SERIALE

Molti dei comandi AT disponibili, oltre alla sintassi
e al numero di parametri ad essi associati, preve-
dono delle risposte più o meno articolate, formate
da diversi parametri, che necessitano del codice per
essere decodificate e rese disponibili all'utente finale
per l'implementazione nella propria applicazione.
Quindi è fondamentale poter studiare il comporta-

mento dei comandi AT che si vuole utilizzare prima
di implementarli nelle proprie applicazioni o librerie.
Inoltre è bene sapere che per raggiungere alcuni
obiettivi, come ad esempio la gestione delle connes-
sioni GPRS, è necessario impartire diversi comandi
AT per attivare la connessione dati. Aggiungiamo
che oltre alla sintassi è necessario anche studiare e
capire la giusta sequenza di comandi AT da inviare
perché alcuni comandi per essere recepiti e accettati
necessitano l'invio preventivo di altri comandi AT
prima di essi e nella giusta sequenza.

Un esempio di questo lo si ha quando si tenta la
connessione a un APN tramite il comando AT+CSTT
in una connessione dati di tipo TCP/UDP. Prima
di inviare il comando AT di connessione allo APN
si deve prima inviare il comando AT+CIPRXGET
(*Get Data From Network Manually*) se non si vuole
vedere ritornare il codice di errore "3", ovvero
operazione non permessa, inviando il comando
AT+CIPSTART.

Uno strumento utile allo scopo è l'utilizzo di sof-

Descrizione	Jumper comunicazione seriale e spia										
	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11
Arduino impostazione jumper											
GSM → RXD1 e TXD1 SW (No spia PC)	X	X	X	2-3	2-3	1-2	1-2	X	X	X	X
GSM → RXD1 e TXD1 SW (Si spia PC)	1-2	1-2	1-2	1-2	1-2	1-2	1-2	X	X	X	X
GSM → RXD1 e TXD1 HW (No spia PC)	X	X	X	2-3	2-3	2-3	2-3	X	X	X	X
GSM → RXD1 e TXD1 HW (Si spia PC)	1-2	1-2	1-2	1-2	1-2	2-3	2-3	X	X	X	X
GPS → RXD2 e TXD2 SW (No spia PC)	X	X	X	2-3	2-3	X	X	1-2	1-2	X	X
GPS → RXD2 e TXD2 SW (Si spia PC)	2-3	2-3	2-3	1-2	1-2	X	X	1-2	1-2	X	X
GPS → RXD2 e TXD2 HW (No spia PC)	X	X	X	2-3	2-3	X	X	2-3	2-3	X	X
GPS → RXD2 e TXD2 HW (Si spia PC)	2-3	2-3	2-3	1-2	1-2	X	X	2-3	2-3	X	X
RaspberryPi impostazione jumper											
GSM → RX e TX (No spia PC)	X	X	X	2-3	2-3	X	X	X	X	1-2	1-2
GSM → RX e TX (Si spia PC)	1-2	1-2	1-2	1-2	1-2	X	X	X	X	1-2	1-2
GPS → RX e TX (No spia PC)	X	X	X	2-3	2-3	X	X	X	X	2-3	2-3
GPS → RX e TX (Si spia PC)	2-3	2-3	2-3	1-2	1-2	X	X	X	X	2-3	2-3
Comandi AT da PC impostazione jumper											
GSM	X	1-2	1-2	2-3	2-3	X	X	X	X	X	X
GPS	X	2-3	2-3	2-3	2-3	X	X	X	X	X	X

Tabella 1

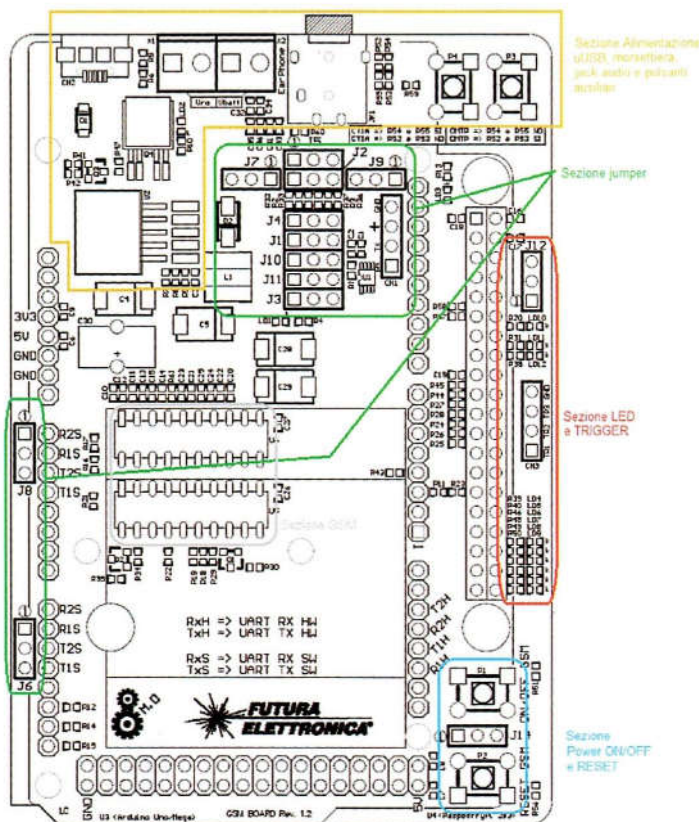


Fig. 9 - Suddivisione del PCB in aree funzionali.

tware per l'analisi e la simulazione dei protocolli di comunicazione seriale. Riportiamo di seguito alcuni esempi nei quali mostreremo delle sezioni di spia dei comandi AT scambiati tra le schede Arduino e il modulo GSM e l'invio dei comandi AT da PC per la configurazione del modulo GSM per lavorare in modalità GPRS.

La Fig. 10 mostra il monitor seriale dell'IDE Arduino durante le fasi di inizializzazione del modulo GSM: in questo caso è un SIMCOM SIM800C. All'avvio del monitor seriale, Arduino viene resettata e viene dato inizio alla procedura di inizializzazione dei moduli GSM, accensione compresa; tuttavia, dato che si partiva da una condizione di modulo GSM già acceso, l'impulso inviato al motore in realtà lo spegne e questo si deduce dalla ricezione della stringa "NORMAL POWER DOWN" (vedere sezione cerchiata in blu). L'algoritmo di inizializzazione è in grado di capire queste situazioni e automaticamente provvede alla riaccensione del modulo GSM, questo si capisce dalla ricezione della stringa "RDY" (vedere sezione cerchiata in rosso). Segue poi una serie di comandi AT per inizializzare il motore GSM tra cui l'invio del codice PIN. Le informazioni che si

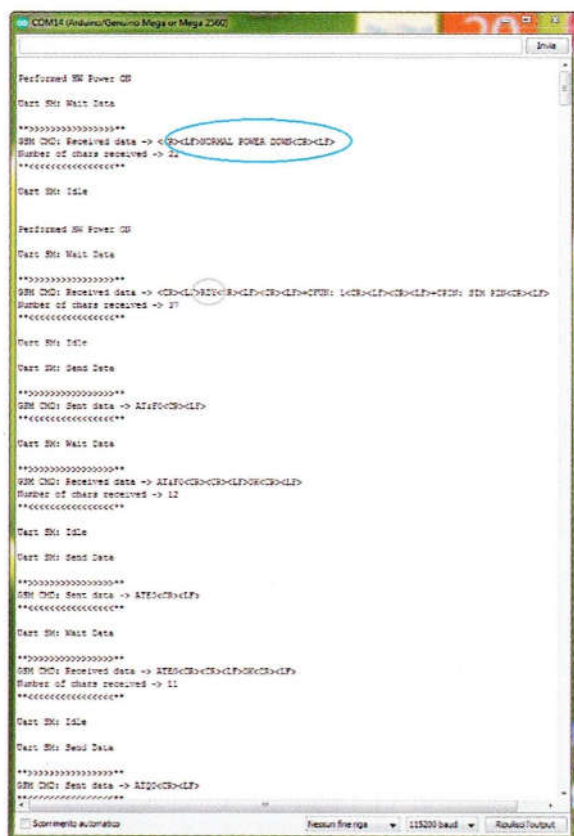


Fig. 10 - Monitor seriale con il modulo GSM in funzione.



Fig. 11
Spegnimento
del modulo
GSM
visto da
monitor
seriale.

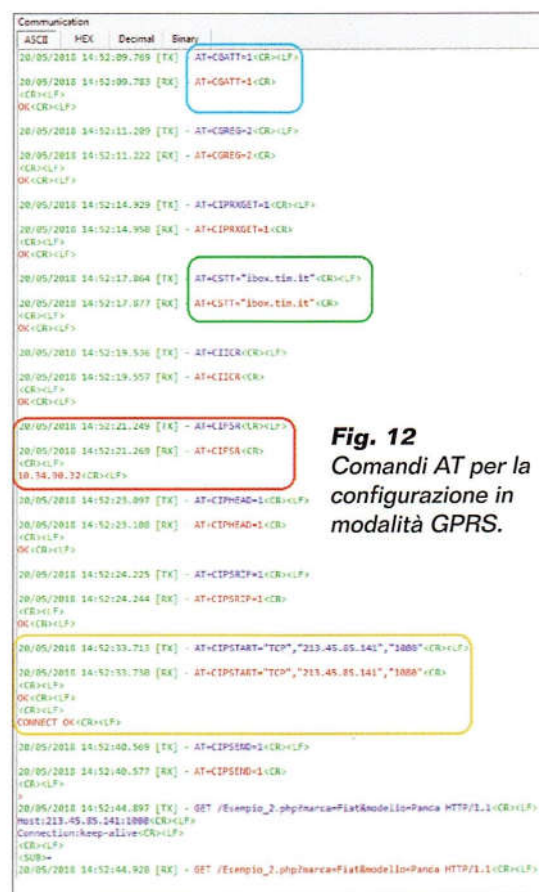


Fig. 12
Comandi AT per la
configurazione in
modalità GPRS.

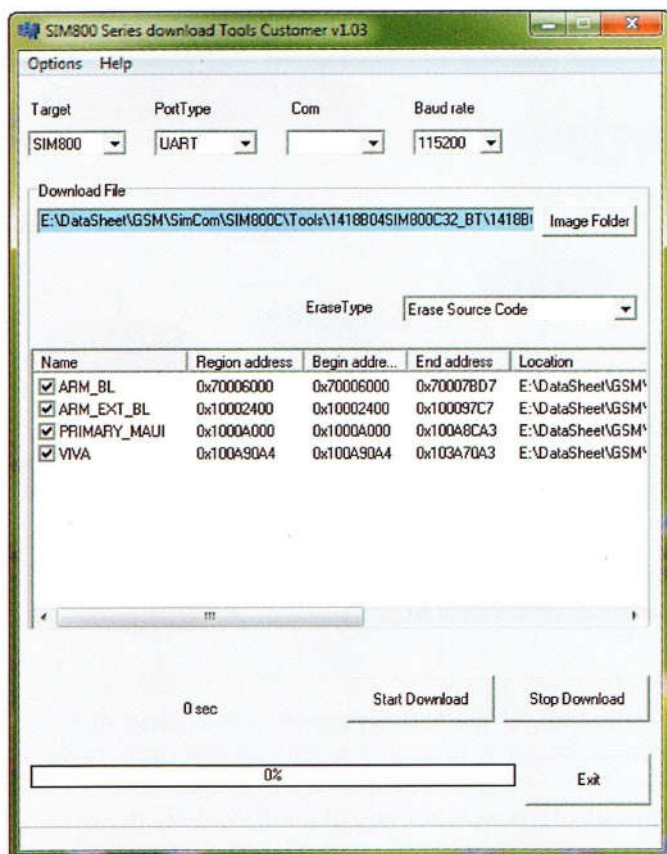


Fig. 13 - Finestra di aggiornamento.

vedono nel monitor seriale dello IDE possono essere abilitate/disabilitate via codice, tali informazioni sono molto utili durante le fasi di debug del codice. Tuttavia per avere una visione reale delle informazioni scambiate tra motore e schede Arduino/Raspberry Pi si deve usare una spia seriale in hardware coadiuvata da un apposito software di analisi.

La Fig. 11 evidenzia, cerchiato in blu, lo spegnimento del modulo GSM da parte della libreria e la successiva riaccensione (in rosso). Cerchiati in verde vedete i comandi AT inviati per verificare se è necessario inviare il codice PIN per attivare la SIM, quanti tentativi abbiamo a disposizione per l'invio del codice PIN e l'avvenuta attivazione della SIM dopo l'invio del codice stesso.

La Fig. 12 invece mostra l'invio dei comandi AT da PC per la configurazione del modulo GSM in modalità GPRS, in particolare durante una connessione dati verso un server tramite protocollo TCP; vedete evidenziato in blu il comando "AT+CGATT=1" usato per connettersi al servizio GPRS, in verde il comando "AT+CSTT="ibox.tim.it" usato per la connessione al proprio APN (GPRS Access Point Name), in rosso il comando "AT+CIFSR" che ritorna l'indirizzo IP assegnato dal GPRS, in giallo il comando "AT+CIPSTART" per la connessione, tramite protocollo TCP o UDP, a un server remoto e relativa porta.

Segue una serie di comandi per l'invio dei dati in particolare sfruttando del codice PHP presente sul server. Quanto appena mostrato evidenzia l'efficacia nell'invio dei comandi AT da PC per lo studio dei passi da seguire per configurare correttamente una connessione dati verso un server remoto. Una volta individuato il giusto flusso di comandi AT si può pensare di implementarli in codice nella propria libreria e relativo sketch.

AGGIORNAMENTO MODULI GSM

Con la connessione diretta al PC è possibile aggiornare il firmware dei moduli GSM; come esempio vediamo come aggiornare il firmware del SimCom SIM800C. Per prima cosa è necessario scaricare il tool, scaricabile ad esempio, dal seguente link:

www.dropbox.com/s/kn73rhcz8v72jhb/SIM800_Series_download_Tools_Customer_v1.06.rar?dl=0.

Non è necessario installare il software in quanto per utilizzarlo è sufficiente eseguire, in modalità amministratore, il file "SIM800_Series_download_Tools_Customer.exe" il quale aprirà la finestra di lavoro in Fig. 13, dove possiamo selezionare una serie di parametri tra cui il **Target** (nel nostro caso, SIM800C), l'interfaccia di comunicazione (**PortType2**, che in questo caso è una UART), la **"Com"** cui si è collegati e infine il **Baudrate**. La **Com** usata nell'esempio è la COM15 e il baud-rate impostato è 115.200.

Il passo successivo consiste nel caricare il firmware, che nel caso dei SimCom è un file a estensione .cfg. L'aggiornamento conviene eseguirlo con il solo shield, perché è indispensabile che il GSM sia spento. Cliccate su "Start Download" e solo dopo accendere il motore GSM tenendo premuto il pulsante

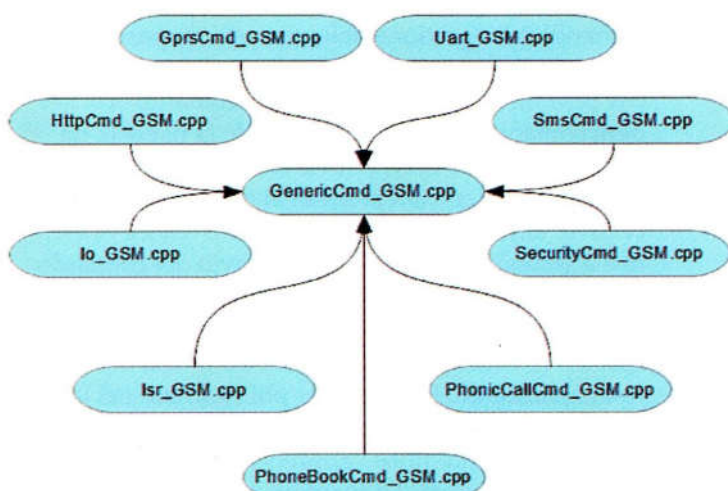


Fig. 14 - Struttura dei file libreria.

di accensione P1 per più di 4 secondi o, se preferite, inserendo il jumper J14 nella posizione 1-2 fino a che non parte la procedura di aggiornamento. L'aggiornamento del firmware avviene in tre step.

LIBRERIA GSM PER LE SCHEDE ARDUINO

L'architettura della libreria si basa su una serie di file in C++ e relativi file .h con le definizioni delle variabili, costanti, struttura dati ecc. Il file principale, contenente anche la procedura di accensione e inizializzazione del modulo GSM, serve da base per gli altri file che compongono la libreria (Fig. 14). Ci sono diversi file di contorno a quello principale tra cui:

- gestione delle connessioni UART "Uart_GSM.cpp"
- gestione degli I/O "Io_GSM.cpp"
- gestione degli interrupt utilizzati dalla libreria "Isr_GSM.cpp"
- gestione degli SMS "SmsCard_GSM.cpp"
- gestione chiamate foniche "PhonicCallCmd_GSM.cpp"
- gestione funzione di sicurezza "SecurityCmd_GSM.cpp"
- gestione rubrica telefonica "PhoneBookCmd_GSM.cpp"
- gestione connessione GPRS "GprsCmd_GSM.cpp"
- gestione comandi HTTP "HttpCmd_GSM.cpp".

I file contengono il codice per la gestione di una serie di comandi AT sia per l'invio dei comandi, con i relativi parametri, sia per la decodifica delle risposte ricevute dal GSM compresi i codici di errore. Attualmente sono gestiti soltanto alcuni comandi AT, ma siccome la libreria è in continuo sviluppo verranno aggiunti sempre più comandi. Stesso discorso per la gestione dei codici di errore.

La Fig. 15 propone la macchina a stati per l'invio dei comandi AT e gestione della risposta da parte del GSM: si inizia con l'inizializzazione del motore GSM per poi entrare in un loop di invio comandi AT. Per ogni comando AT inviato si attende la risposta da parte del motore GSM e se ne decodifica il contenuto. Se la risposta è OK si passa al comando AT successivo, altrimenti il sistema ritorna il codice di errore e a seconda dei casi ritenterà l'invio del comando AT fino a un massimo di tre volte. Se il comando AT non va a buon fine, nemmeno dopo tre tentativi, il sistema resetta il motore GSM e ricomincia da capo re-inizializzando lo stesso per poi ritornare nel loop di invio comandi AT.

Questo è a grandi linee l'approccio che c'è alla base della nostra libreria. Le funzioni per la gestione dei comandi AT che non vengono usate non occupano spazio di memoria Flash, tuttavia per ogni categoria

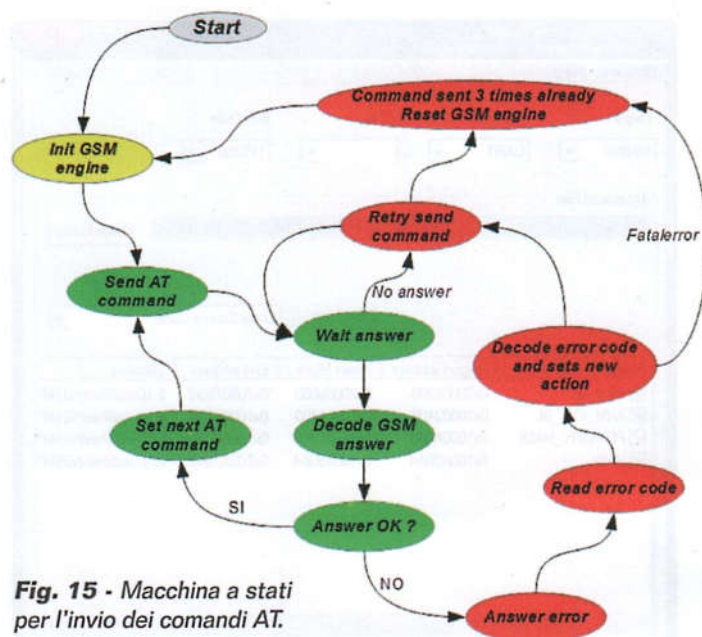


Fig. 15 - Macchina a stati per l'invio dei comandi AT.

di comandi AT esiste una propria sezione di gestione delle risposte, la quale è gestita da una opportuna macchina a stati. In questo caso, se abilitate con apposito #Define, le routine di gestione delle risposte vengono inserite nella compilazione del codice macchina e quindi occupano spazio in Flash. Il tutto sta nel cercare il giusto equilibrio tra funzioni necessarie alle proprie applicazioni e spazio in memoria occupato dalla libreria.

La libreria supporta le schede Arduino Uno e Fishino Uno, Arduino Mega 2560 o Fishino Mega 2560. In futuro non escludiamo di estendere la compatibilità alle Arduino più evolute, ovvero quelle con microcontrollore a 32 bit. Ciò detto, concludiamo qui questa prima puntata.



per il MATERIALE

Lo shield universale GSM (cod. WWGSMESHIELD) è in vendita presso Futura Elettronica al prezzo di Euro 54,90. Viene fornito montato ad esclusione degli strip line (compresi) che vanno saldati manualmente. Il prezzo è comprensivo di IVA.

Lo shield si può interfacciare con uno dei seguenti moduli GSM: interfaccia con M95 (cod. FT1128M, Euro 39,00), modulo cellulare miniaturizzato con SIM800 (cod. FT1308M, Euro 29,00) e modulo cellulare GSM/GPS con SIM928A (cod. FT1178M, Euro 59,00).

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287 - www.futurashop.it

COOPERATIVE SCHEDULER

..... di GUGLIELMO BRAGUGLIA

Conosciamo un semplicissimo Scheduler Cooperativo in grado di gestire i task da far eseguire alle board Arduino e compatibili.

Molto spesso gli utenti Arduino si trovano ad affrontare il problema di dover eseguire, in tempi ben definiti e quindi in maniera pianificata, determinate azioni; la soluzione che verrebbe più immediata è l'utilizzo di una funzione insita in Arduino, che è la `millis()` e che possiamo considerare un contatore che ci comunica quanti millisecondi sono trascorsi da quando la board Arduino è stata avviata, ovvero dall'ultimo reset. Purtroppo, se è vero che `millis()` è lì pronta da utilizzare, è vero altrettanto che non è così facile come dirlo, perché occorre fare i conti con il corretto utilizzo e l'applicazione pratica di detta funzione. Per chi avesse trovato ostico e tedioso districarsi tra le difficoltà di applicazione di `millis()`, abbiamo pensato a una soluzione semplice ed elegante per rispondere all'esigenza di programmare temporalmente l'esecuzione di determinate azioni: si tratta dell'implementazione di un semplicissimo "scheduler cooperativo" che richiede pochissime risorse e che può essere tranquillamente utilizzato anche su Arduino UNO, che notoriamente dispone di poca memoria.

COS'È UNO SCHEDULER COOPERATIVO?

Come dice il nome stesso, uno scheduler cooperativo è un programma che permette la messa in esecuzione, in modo "collaborativo" di diverse

parti di programma immaginabili come delle semplici funzioni. Pensiamo al classico caso in cui si deve ogni T1 secondi leggere un sensore di temperatura, ogni T2 secondi leggere un sensore di umidità, ogni T3 secondi leggere un sensore di pressione, ogni T4 secondi leggere un RTC ed ogni T5 secondi aggiornare un LCD con i dati letti nelle varie fasi precedenti. Una strada è fare tutto nel loop(), decidendo quando fare le varie cose con l'utilizzo di millis(); un'altra strada è fare la stessa cosa appoggiandosi su un meccanismo che usa lui correttamente la funzione millis() e che toglie al programmatore il compito di effettuare tutti i vari controlli temporali ... Il più semplice di questi meccanismi è un piccolo programma che viene chiamato "scheduler cooperativo" e che non richiede il diretto utilizzo di meccanismi hardware come, ad esempio, timer programmabili o interrupt, cui viene detto quali funzioni richiamare e con che periodicità richiamarle lasciando al programmatore il solo compito di scrivere delle piccole funzioni (... che, impropriamente, possiamo chiamare Task) che svolgono i vari compiti, senza preoccuparsi di come temporalmente richiamarle (Fig. 1).

Ovviamente, affinché uno "Scheduler Cooperativo" funzioni correttamente occorre però rispettare alcune semplici regole:

1. la somma delle durate di tutte le singole funzioni deve essere inferiore all'intervallo minimo richiesto allo scheduler per l'attivazione di una qualsiasi delle funzioni;
2. diretta conseguenza del punto 1 è che, ovviamente, anche nessuna singola funzione deve durare più dell'intervallo minimo richiesto allo scheduler per l'attivazione di una qualsiasi delle funzioni; ma questo è comunque valido sia che si utilizzi direttamente la funzione millis(), sia se ci si semplifichi la vita usando uno "scheduler cooperativo".

Notate che nel caso non sia possibile rispettare queste regole, la cosa si complica ed occorre utilizzare un vero scheduler con tanto di gestione di interrupt, gestione di priorità e gestione del meccanismo di "prelazione" (*preemptive*), come, ad esempio, può essere FreeRTOS™, che sarà però argomento di un articolo dedicato di futura pubblicazione.

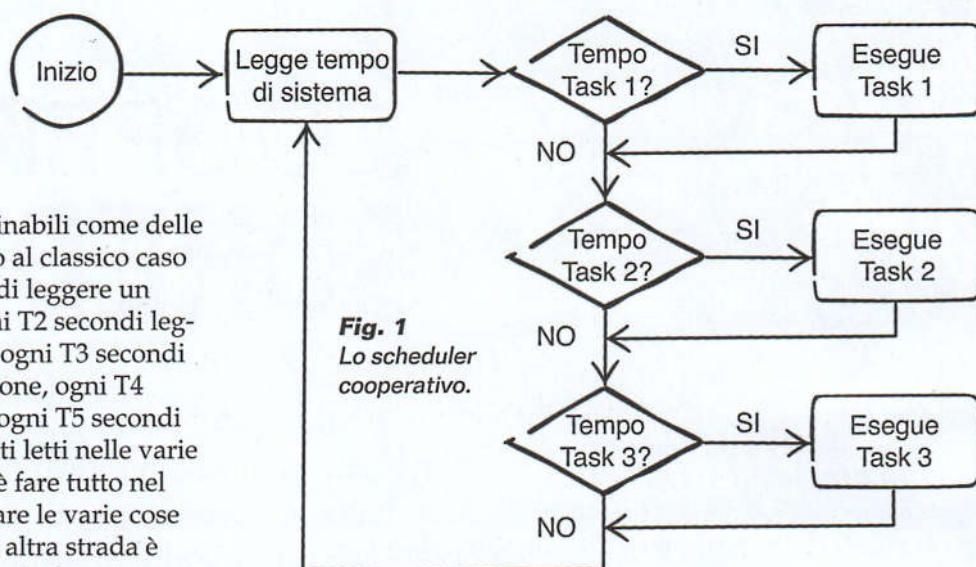


Fig. 1
Lo scheduler cooperativo.

Riprendendo l'esempio proposto qualche paragrafo indietro, avremo quindi:

1. una semplice funzione che ha il solo compito di leggere il sensore di temperatura e di mettere il valore letto in una variabile.
2. una semplice funzione che ha il solo compito di leggere il sensore di umidità e di mettere il valore letto in una variabile.
3. una semplice funzione che ha il solo compito di leggere il sensore di pressione e di mettere il valore letto in una variabile.
4. una semplice funzione che ha il solo compito di leggere il RTC e di mettere il valore letto in una variabile.
5. una semplice funzione che ha il solo compito di prendere le variabili riempite dalle precedenti funzioni e visualizzarle su un LCD.

Si tratta di cinque funzioni, indipendenti tra di loro che iniziano, fanno (... in pochi millisecondi) ciò che devono fare e terminano.

Compito del programmatore sarà solo dire allo "scheduler cooperativo" che ci sono 5 funzioni (Tasks) e con che intervallo di tempo occorre richiamarle.

Sempre facendo riferimento all'esempio, si potrebbe decidere di leggere la temperatura ogni 5 minuti (T1 = 30000 millisecondi), di leggere l'umidità anche essa ogni 5 minuti (T2 = 30000 millisecondi), di leggere la pressione ogni 10 minuti (T3 = 60000 millisecondi), di leggere il RTC ogni secondo (T4 = 1000 millisecondi) e di aggiornare il LCD anche esso ogni secondo (T5 = 1000 millisecondi) così da avere l'ora visualizzata sempre aggiornata, comunicando questi tempi allo "scheduler cooperativo" che se ne

Listato 1

```
typedef struct {
    uint32_t Interval; // Defines how often a task will run
    uint32_t LastTick; // Stores the last tick task was ran
    uint32_t TaskSpan; // Task duration in usec
    boolean CanRun; // Indicate if the task can run
    boolean AsRun; // Indicate if the task has run at least once
    void (*Func)(void); // Function pointer to the task
} Task_t;
```

occuperà direttamente togliendo l'incombenza al programmatore.

IMPLEMENTIAMO LO "SCHEDULER COOPERATIVO"

Vediamo quindi ora come scrivere un modulo, genericamente riutilizzabile, adatto ad implementare un semplicissimo "scheduler cooperativo" in ambiente Arduino (... *ma facilmente trasportabile in qualsiasi ambiente*).

Prima di tutto dobbiamo costruire una "struttura" dove memorizzeremo le informazioni relative ad ogni singolo task (*le nostre singole funzioni*).

Pertanto definiremo la parte di codice che vedete nel Listato 1.

In questo codice:

- "Interval": è una variabile da 32 bit che definirà ogni quanti millisecondi il task deve essere messo in esecuzione (... *la nostra funzione chiamata*);
- "LastTick": è una variabile da 32 bit dove si memorizzerà il "tempo" (il valore di millis()) dell'ultima volta che il task è stato chiamato;
- "TaskSpan": è una variabile da 32 bit dove verrà memorizzato quanti microsecondi il nostro task ha impiegato ad essere eseguito; tale valore NON è necessario allo scheduler di per se, ma è utile, in una fase di debug, per sapere quanto tempo dura l'esecuzione di ogni nostra singola funzione;
- "CanRun": è una variabile booleana ed indica se il task può essere eseguito o meno (*ad esempio per sospendere, da programma, l'esecuzione di un task per un determinato periodo o per sempre*);
- "AsRun": è una variabile booleana utilizzata dallo scheduler per sapere se il task è la prima volta che viene eseguito o è già stato eseguito dallo scheduler almeno una volta;
- (*Func) (void): è il puntatore alla funzione che implementa il singolo task (*ovvero la nostra singola funzione che esegue un determinato compito*);

Dichiareremo poi un puntatore ad un blocco di tipo "Task_t" che abbiamo appena definito:

```
static Task_t *Task_ptr;
```

e che utilizzeremo per memorizzare il puntatore ad un blocco di memoria che allocheremo di dimen-

sioni tali da poter contenere una struttura di tipo Task_t per ogni nostro singolo task (*in pratica un array di elementi di tipo Task_t*).

Ci occorreranno poi alcune semplici variabili che useremo per:

```
static uint8_t fInit = false;
// flag to indicate successfully initialization
static uint8_t NumberOfTasks = 0;
// maximum number of tasks
static uint8_t fIdle = false;
// flag to indicate "nothing to do"
```

- "fInit": è una variabile da 1 byte che useremo per ricordarci se la fase di inizializzazione è stata effettuata o meno (... *fase in cui deve venire creato l'array di Task_t ed inizializzato*).
- "NumberOfTasks": è una variabile da 1 byte che useremo per ricordare il numero di task (*quante sono le nostre singole funzioni*).
- "fIdle": è una variabile da 1 byte che ci servirà per indicare se, nel ciclo di scheduling, è stato chiamato qualche task o non si è fatto nulla.

Siamo ora pronti per definire le funzioni che ci occorreranno per gestire il nostro semplice "Scheduler Cooperativo". Per prima cosa definiamo una funzione di inizializzazione che andrà chiamata una ed una sola volta per inizializzare il nostro scheduler;

Listato 2

```
uint8_t csInit(uint8_t NTasks) {
    uint8_t idx;
    uint16_t memSz;
    //
    if (fInit) return 0; // can't initialize two times
    //
    memSz = NTasks * sizeof(Task_t);
    Task_ptr = (Task_t *)malloc(memSz);
    if (Task_ptr != NULL) {
        fInit = true;
        NumberOfTasks = NTasks;
        for (idx = 0; idx < NumberOfTasks; idx++) {
            Task_ptr[idx].Interval = 0;
            Task_ptr[idx].LastTick = 0;
            Task_ptr[idx].TaskSpan = 0;
            Task_ptr[idx].CanRun = false;
            Task_ptr[idx].AsRun = false;
            Task_ptr[idx].Func = NULL;
        }
        #ifdef csDEBUG
        Serial.print(F("Memory allocated for TCBs: "));
        Serial.print(memSz);
        Serial.println(F(" bytes"));
        #endif
        return 1;
    }
    else return 0;
}
```


csSched.h initialization function

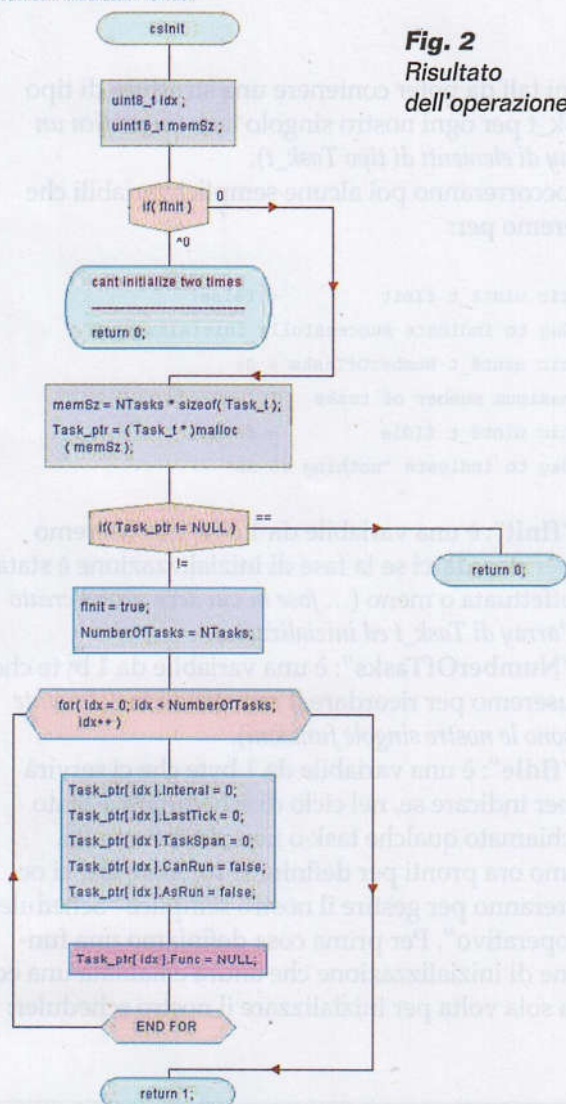


Fig. 2
Risultato
dell'operazione.

csSched.h task adding function

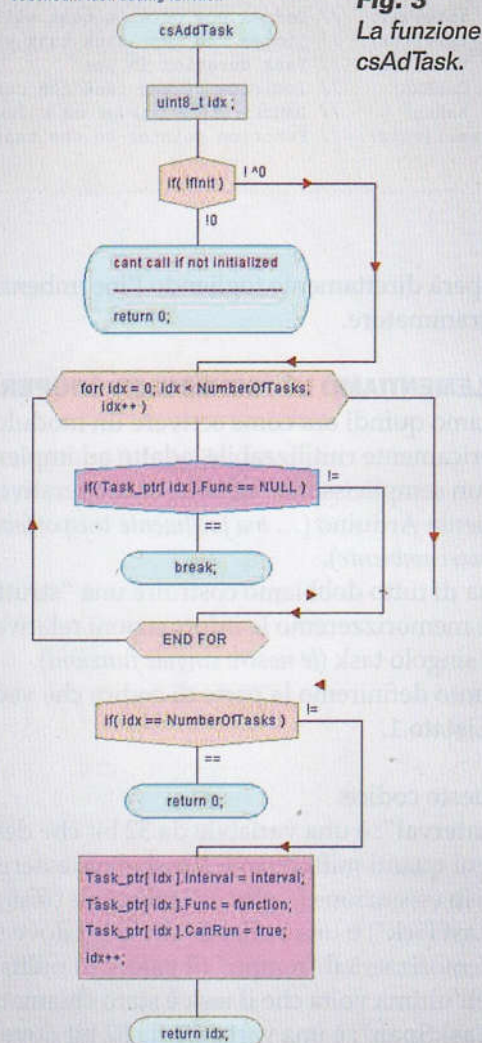


Fig. 3
La funzione
csAddTask.

tale funzione è riportata nel Listato 2.

In tale funzione, come prima cosa controlliamo di venire chiamati una ed una sola volta (con l'utilizzo della variabile *finit*) e poi, in funzione del numero massimo di task che il chiamante prevede (parametro *NTasks*), allochiamo lo spazio di memoria per *NTasks* volte la struttura *Task_t* (una per ogni task). Se l'operazione riesce e la memoria viene correttamente allocata, la funzione ritorna "true" (1), mentre se c'è un errore essa ritorna "false" (0). La Fig. 2 mostra la funzione *csInit*. Creato lo spazio di memoria dove lo scheduler può conservare i dati di ogni singolo task, occorre un'ulteriore funzione che permetta di specificare i vari task (le nostre varie funzioni) e l'intervallo di tempo con cui devono essere richiamate; tale funzione è riportata nel Listato 3. Anche questa funzione, come prima cosa, controlla che la fase di inizializzazione sia stata eseguita correttamente, dopo di che, cerca tra i vari blocchi

Task_t che sono stati precedentemente allocati, un blocco libero dove inserire i dati relativi all'intervallo di tempo con cui il task deve essere chiamato (parametro *interval*) ed al nome della funzione che implementa tale task (parametro *(*function)()*). Se l'operazione ha successo e i dati del nuovo task sono stati memorizzati, la funzione ritorna un numero che indica il "numero identificativo" del task aggiunto (parte da 1 a crescere); se invece l'operazione non ha avuto successo ritorna 0 (zero) ad indicare la condizione di errore (Fig. 3).

Ultima funzione strettamente necessaria (ne aggiungeremo alcune altre dopo, ma solo a fine di completezza dello scheduler) è quella che dà il via al vero e proprio scheduler e che quindi prende il controllo del programma; la vedete nel Listato 4.

Come le precedenti, anche questa funzione controlla che la fase di inizializzazione sia stata fatta, ma, contrariamente alle altre che tornavano al chiama-

te un'indicazione di riuscita o meno dell'operazione, questa, che ha il compito di prendere il controllo e di non tornare nulla al chiamante, in caso di errore (quindi in caso di mancata inizializzazione), blocca l'esecuzione del programma facendolo entrare in un ciclo chiuso dal quale non si esce più. Se tutto è a posto e si può iniziare a fare eseguire i vari task (le nostre varie funzioni), lo scheduler prende il controllo e, agli intervalli prefissati, esegue le varie funzioni associate ai vari task (Fig. 4).

Elenchiamo adesso alcuni punti da notare in tale funzione.

1. Il parametro booleano "fStartAtTime" condiziona solo la prima esecuzione in "immediata", o "al tempo previsto", per tutti i tasks; se esso è posto a "true" il campo Task_ptr[TaskIndex].LastTick del blocco identificativo del singolo task viene inizializzato al valore attuale di millis(), il che comporterà la prima esecuzione di ogni singolo task, solo allo scadere dell'intervallo previsto. Se invece esso è posto a "false", ogni singolo task viene immediatamente eseguito una volta, dopo di che il campo Task_ptr[TaskIndex].LastTick del blocco identificativo del singolo task, viene inizializzato al valore attuale di millis() per permettere le successive esecuzioni allo scadere dell'intervallo previsto.
2. Il campo TaskSpan viene calcolato prendendo i micros() prima della messa in esecuzione del singolo task (la singola funzione) e subito dopo, facendo quindi la differenza per calcolare quanto è durato il singolo task.

Listato 3

```
uint8_t csAddTask(uint32_t interval, void(*function)()) {
    uint8_t idx;
    //
    if (!fInit) return 0; // can't call if not initialized
    //
    for (idx = 0; idx < NumberOfTasks; idx++) {
        if (Task_ptr[idx].Func == NULL) break;
    }
    if (idx == NumberOfTasks) return 0;
    //
    Task_ptr[idx].Interval = interval;
    Task_ptr[idx].Func = function;
    Task_ptr[idx].CanRun = true;
    //
    idx++;
    #ifdef CSDEBUG
        Serial.print(F("Added task number "));
        Serial.println(idx);
    #endif
    return idx;
}
```

csSched.h Cooperative Scheduler function

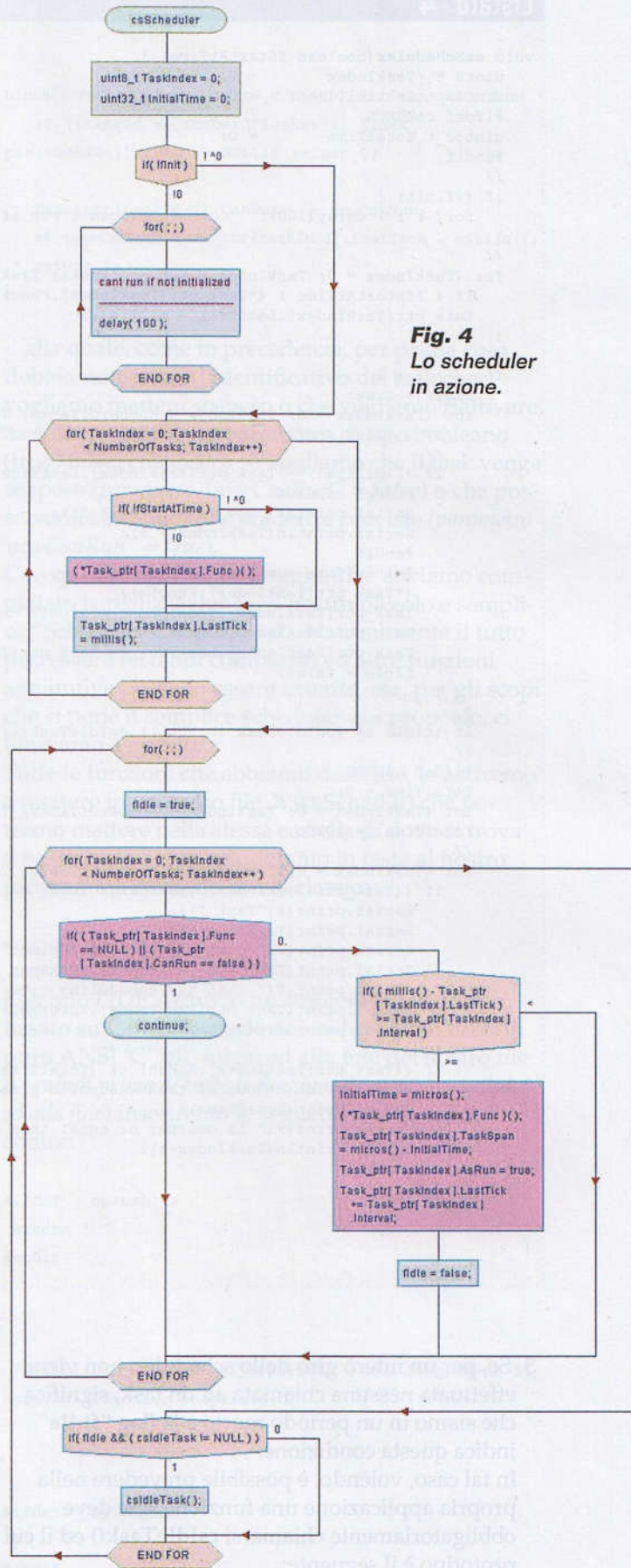


Fig. 4
Lo scheduler
in azione.

Listato 4

```

void csScheduler(boolean fStartAtTime) {
    uint8_t TaskIndex      = 0;
    uint32_t InitialTime    = 0;
    #ifdef csDEBUG
    uint32_t TotalTime      = 0;
    #endif
    //
    if (!fInit) {
        for(;;) delay(100);          // can't run if not initialized
    }
    //
    for (TaskIndex = 0; TaskIndex < NumberOfTasks; TaskIndex++) {
        if ( !fStartAtTime ) (*Task_ptr[TaskIndex].Func)();
        Task_ptr[TaskIndex].LastTick = millis();
    }
    //
    for ( ; ; ) {
        fIdle = true;
        for (TaskIndex = 0; TaskIndex < NumberOfTasks; TaskIndex++) {
            if ( (Task_ptr[TaskIndex].Func == NULL) || (Task_ptr[TaskIndex].CanRun == false) ) continue;
            //
            if ( (millis() - Task_ptr[TaskIndex].LastTick) >= Task_ptr[TaskIndex].Interval ) {
                #ifdef csDEBUG
                Serial.print(F("Now running task "));
                Serial.println(TaskIndex + 1);
                #endif
                InitialTime = micros();
                (*Task_ptr[TaskIndex].Func)();
                Task_ptr[TaskIndex].TaskSpan = micros() - InitialTime;
                Task_ptr[TaskIndex].AsRun = true;
                Task_ptr[TaskIndex].LastTick += Task_ptr[TaskIndex].Interval;
                fIdle = false;
            }
        }
        if (fIdle && (csIdleTask != NULL)) csIdleTask();
        //
        #ifdef csDEBUG
        TotalTime = 0;
        for (TaskIndex = 0; TaskIndex < NumberOfTasks; TaskIndex++) TotalTime += Task_ptr[TaskIndex].TaskSpan;
        TotalTime /= 1000;
        //
        for (TaskIndex = 0; TaskIndex < NumberOfTasks; TaskIndex++) {
            if ((Task_ptr[TaskIndex].TaskSpan / 1000) >= Task_ptr[TaskIndex].Interval) {
                Serial.print(F("Task "));
                Serial.print(TaskIndex+1);
                Serial.print(F(" uses more or equal time: "));
                Serial.print(Task_ptr[TaskIndex].TaskSpan / 1000);
                Serial.print(F(" than his scheduling time "));
                Serial.print(Task_ptr[TaskIndex].Interval);
                Serial.println(F(""));
            }
            if ((Task_ptr[TaskIndex].AsRun) && (TotalTime >= Task_ptr[TaskIndex].Interval)) {
                Serial.print(F("Sum of task execution times: "));
                Serial.print(TotalTime);
                Serial.print(F(" is gearter or equal than scheduling time of task "));
                Serial.println(TaskIndex+1);
            }
        }
        #endif
    }
}

```

3. Se, per un intero giro dello scheduler, non viene effettuata nessuna chiamata ad un task, significa che siamo in un periodo morto e la flag "fIdle" indica questa condizione. In tal caso, volendo, è possibile prevedere nella propria applicazione una funzione, che deve obbligatoriamente chiamarsi `csIdleTask()` ed il cui prototipo è il seguente:

```
void csIdleTask(void);
```

Tale funzione, che deve obbligatoriamente essere la più veloce possibile, può essere, ad esempio, usata a fini statistici per calcolare la percentuale di occupazione della MCU o altre funzionalità del genere ... operazioni da fare quindi durante i tempi morti e di brevissima durata.

4. Se, in testa al codice, viene inserita la **#define csDEBUG**, vengono attivate una serie di segnalazioni che aiutano a fare il debug di eventuali problemi. Dato che le informazioni vengono stampate su "Serial", occorrerà che il programma principale, prima di chiamare le funzioni del nostro semplice "Scheduler Cooperativo", abbia provveduto ad inizializzare tale porta (*Serial.begin()*) e che sia aperto il monitor seriale dove vedere gli eventuali messaggi.

Una volta che il programma è correttamente funzionante, vi rammentiamo di rimuovere la suddetta **#define** così da ridurre i tempi di esecuzione e l'occupazione di memoria.

COMPLETIAMO LO SCHEDULER

Per completare il nostro semplice "Scheduler Cooperativo", aggiungiamo un paio di piccole funzioni che ci possono tornare utili durante lo sviluppo delle nostre applicazioni.

La prima funzione che aggiungiamo ci permette di variare il tempo con cui viene richiamato un task (*una delle nostre funzioni*). Abbiamo visto che questo valore viene fornito, in fase di aggiunta del task alla lista dei task da eseguire, attraverso il parametro "interval" della funzione *csAddTask()*.

Per poterlo, poi, modificare durante l'esecuzione del programma, possiamo utilizzare questa semplice funzione:

```
uint8_t csChangeTaskInterval(uint8_t taskId, uint32_t
newInterval) {
    if ((taskId >= NumberOfTasks) || (Task_
ptr[taskId-1].Func == NULL)) return 0;
    //
    Task_ptr[taskId-1].Interval = newInterval;
    return 1;
}
```

... alla quale dobbiamo passare l'identificativo del task di cui vogliamo cambiare la periodicità "taskId" (*identificativo che ci è stato ritornato dalla succitata funzione csAddTask() al momento dell'aggiunta di un task*) e la nuova periodicità "newInterval", sempre espressa in millisecondi.

Come sempre il successo della modifica è indicato dal valore di ritorno "true" (1), mentre un eventuale errore è indicato dal valore di ritorno "false" (0).

La seconda funzione che aggiungiamo ci permette di mettere in pausa/riattivare un task così che venga ignorato/ripreso dallo scheduler durante il suo ciclo. A tale scopo creiamo un'ulteriore funzione:

```
uint8_t csChangeCanRun(uint8_t taskId, boolean newCanRun) {
    if ((taskId >= NumberOfTasks) || (Task_
ptr[taskId-1].Func == NULL)) return 0;
    //
    Task_ptr[taskId-1].CanRun = newCanRun;
    if (newCanRun) Task_ptr[taskId-1].LastTick = millis();
    return 1;
}
```

... alla quale, come in precedenza, per prima cosa dobbiamo passare l'identificativo del task che vogliamo mettere sospeso o che vogliamo riattivare, "taskId", e quindi un parametro di tipo booleano (true/false) che indica se vogliamo che il task venga sospeso (*parametro "newCanRun" = false*) o che possa venire eseguito alle scadenze previste (*parametro "newCanRun" = true*).

Con queste due funzioni aggiuntive abbiamo completato la realizzazione del nostro piccolo e semplice "Scheduler Cooperativo". Naturalmente il tutto può essere reso più complesso ed altre funzioni aggiuntive possono essere inserite, ma, per gli scopi che si pone il semplice scheduler qui proposto, ci limitiamo a questo.

Tutte le funzioni che abbiamo descritto, le andremo a mettere in un unico file .h (*csSched.h*) che potremo mettere nella stessa cartella di dove si trova il nostro .ino e che includeremo in testa al nostro programma Arduino con la classica:

```
#include "csSched.h"
```

Essendo l'IDE Arduino un ambiente di sviluppo basato su C++ ed avendo noi sviluppato il tutto in puro ANSI 'C', all'inizio ed alla fine del nostro file .h (*quindi prima di tutte le funzioni del nostro scheduler ed alla fine*) inseriremo le seguenti poche righe di codice:

```
#ifndef __cplusplus
extern "C" {
#endif
```

```
...
// tutte le funzioni qui descritte e che compongono il
nostro "Scheduler cooperativo"
```

```
#ifndef __cplusplus
}
#endif
```



```
#define csDEBUG

#include <Wire.h>
#include <BMP085.h>
#include <DHT.h>
#include <DS3231.h>
#include <PCF8574_HD44780_I2C.h>
#include "csSched.h"

//

// #define DS1307 0x68 // Indirizzo I2C del RTC (DS1307)
// #define LCDDIS 0x27 // Indirizzo I2C del LCD
// #define BMP085 0x77 // Indirizzo I2C del sensore BMP085

#define DHTTYPE DHT22 // Tipo di sensore DHT utilizzato (DHT22)
#define DHTPIN 8 // pin Arduino a cui è collegato il DHT

//

BMP085 bmp; // istanza della classe BM180
DHT dht022(DHTPIN, DHTTYPE); // istanza della classe DHT
PCF8574_HD44780_I2C lcd(LCDDIS, 20, 4); // istanza della classe PCF8574_HD44780_I2C
DS3231 Clock;

int32_t bmp180_Pa; // variabile dove salvare la pressione atmosferica
float bmp180_T; // variabile dove salvare la temperatura
float bmp180_Al; // variabile dove salvare l'altrezza in mt.
float dht022_H;
int ds3231_date[7] = {0,0,0,0,0,0,0}; // array dove salvare i vari elemnti letti dal RTC
bool Century=false;
bool h12;
bool PM;
//

/*byte bcdToDec(byte val) // funzione per la conversione di un dato BCD in decimale
{
    return ( (val/16*10) + (val%16) );
}
*/
void ds3231_Read(void) // funzione per la lettura dei vari registri del RTC
{
    ds3231_date[0] = Clock.getSecond(); // legge i secondi
    ds3231_date[1] = Clock.getMinute(); // legge i minuti
    ds3231_date[2] = Clock.getHour(h12, PM); // legge le ore
    ds3231_date[3] = Clock.getDoW(); // legge il giorno della settimana (1 = domenica, 7 = sabato)
    ds3231_date[4] = Clock.getDate(); // legge il giorno del mese
    ds3231_date[5] = Clock.getMonth(Century); // legge il mese
    ds3231_date[6] = Clock.getYear(); // legge l'anno
}

//

void leggeBMP085(void) { // Primo Task : lettura del BMP085
    bmp180_T = bmp.readTemperature(); // legge la temperatura dal BMP085
    bmp180_Pa = bmp.readPressure(); // legge la pressione dal BMP085
    bmp180_Al = bmp.getAltitude(bmp180_Pa); // correzione dell'altitudine ler la mia localita'
}

void leggeDHT(void) { // Secondo Task: lettura del DHT
    dht022_H = dht022.readHumidity(); // legge l'umidita' dal DHT
    if (isnan(dht022_H)) dht022_H = 0.0; // e verifica che il valore sia numerico
}

void leggeRTC(void) { // Terzo Task : lettura del RTC
    ds3231_Read(); // legge i dati del RTC
}

void scriviSuLCD(void) { // Quarto Task : visualizzazione dei dati sul LCD
    lcd.setCursor(0, 0); // posiziona il cursore a col. 0, rig. 0

    lcd.print("20");
    lcd.print(ds3231_date[6], DEC); // YY
    lcd.print(".");
    if (ds3231_date[5] < 10) lcd.print("0"); // MM
    lcd.print(ds3231_date[5], DEC);
    lcd.print(".");
    if (ds3231_date[4] < 10) lcd.print("0"); // DD
    lcd.print(ds3231_date[4], DEC);
    lcd.print(" ");
    if (ds3231_date[2] < 10) lcd.print("0"); // hh
    lcd.print(ds3231_date[2], DEC);
    lcd.print(":");
    if (ds3231_date[1] < 10) lcd.print("0"); // mm
    lcd.print(ds3231_date[1], DEC);
    lcd.print(":");
    if (ds3231_date[0] < 10) lcd.print("0"); // ss
    lcd.print(ds3231_date[0], DEC);

    lcd.setCursor(0, 1); // posiziona il cursore a col. 0, rig. 1

    lcd.print("T:");
    lcd.print(bmp180_T, 1); // Temperatura
    lcd.print(" ");
    lcd.print((char) 0xDF);
```



```

    lcd.print("C. ");

    lcd.print(" H:");
    lcd.print(dht022_H, 1);
    lcd.print(" %"); // Umidita'

    lcd.setCursor(0, 2); // posiziona il cursore a col. 0, rig. 2

    lcd.print("P:");
    lcd.print((int32_t) bmp180_Pa / 100); // Pressione
    lcd.print(" mmHg. ");

    lcd.setCursor(0, 3); // posiziona il cursore a col. 0, rig. 3

    lcd.print("A:");
    lcd.print((int32_t) bmp180_Al); // Altezza
    lcd.print(" mt.");
}

void hardwareInit (void) { // funzione per l'inizializzazione del hardware
    Serial.begin(115200); // inizializza la porta seriale
    //
    pinMode(LED_BUILTIN, OUTPUT); // inizializza il pin del LED sulla scheda
    digitalWrite(LED_BUILTIN, LOW); // spegne il LED sulla scheda
    //
    lcd.init(); // inizializza il LCD ...
    lcd.backlight(); // ... ne attiva il backlight
    lcd.clear(); // ... pulisce il LCD
    lcd.setCursor(0,0); // ... posiziona il cursore a 0,0
    lcd.print(" Cooperative ");
    lcd.setCursor(0,1);
    lcd.print(" Scheduler");
    lcd.setCursor(0,3);
    lcd.print(" Ready");
    lcd.setCursor(0, 0); // posiziona il cursore a col. 0, rig. 0
    delay(500);

    //
    while(!bmp.begin(BMP085_ULTRA_HIGH_RES))
    {
        Serial.println("Could not find a valid BMP085 or BMP180 sensor, check wiring!");
        delay(500);
        // inizializza il sensore BMP085
    }
    //
    dht022.begin(); // inizializza il sensore DHT
    delay(1000);
    dht022.readHumidity(); // effettua una lettura a vuoto del DHT
    lcd.clear();
}

//
void setup() {
    uint8_t fOK;
    //
    delay(500);
    //
    hardwareInit(); // inizializza l'hardware
    //
    fOK = csInit(4); // inizializza lo "scheduler cooperativo" per 4 task
    if (!fOK) {
        Serial.println(F("Impossibile inizializzare lo scheduler"));
        for ( ; );
    }
    //
    fOK = csAddTask(30000, leggeBMP085); // Aggiunge il primo task: lettura del BMP085 ogni 30 secondi
    if (!fOK) {
        Serial.println(F("Impossibile creare il task 1"));
        for ( ; );
    }
    fOK = csAddTask(60000, leggeDHT); // Aggiunge il secondo task: lettura del DHT ogni 60 secondi
    if (!fOK) {
        Serial.println(F("Impossibile creare il task 2"));
        for ( ; );
    }
    fOK = csAddTask(500, leggeRTC); // Aggiunge il terzo task: lettura del RTC ogni 0.5 secondi
    if (!fOK) {
        Serial.println(F("Impossibile creare il task 3"));
        for ( ; );
    }
    fOK = csAddTask(1000, scriviSuLCD); // Aggiunge il quarto task: scrittura sul LCD
    if (!fOK) {
        Serial.println(F("Impossibile creare il task 4"));
        for ( ; );
    }
    //
    csScheduler(false); // Attiva lo scheduler
}

//
void loop() {
    //
    // Inutile mettere qualche cosa qui, il loop() NON viene mai raggiunto
    //
}

```

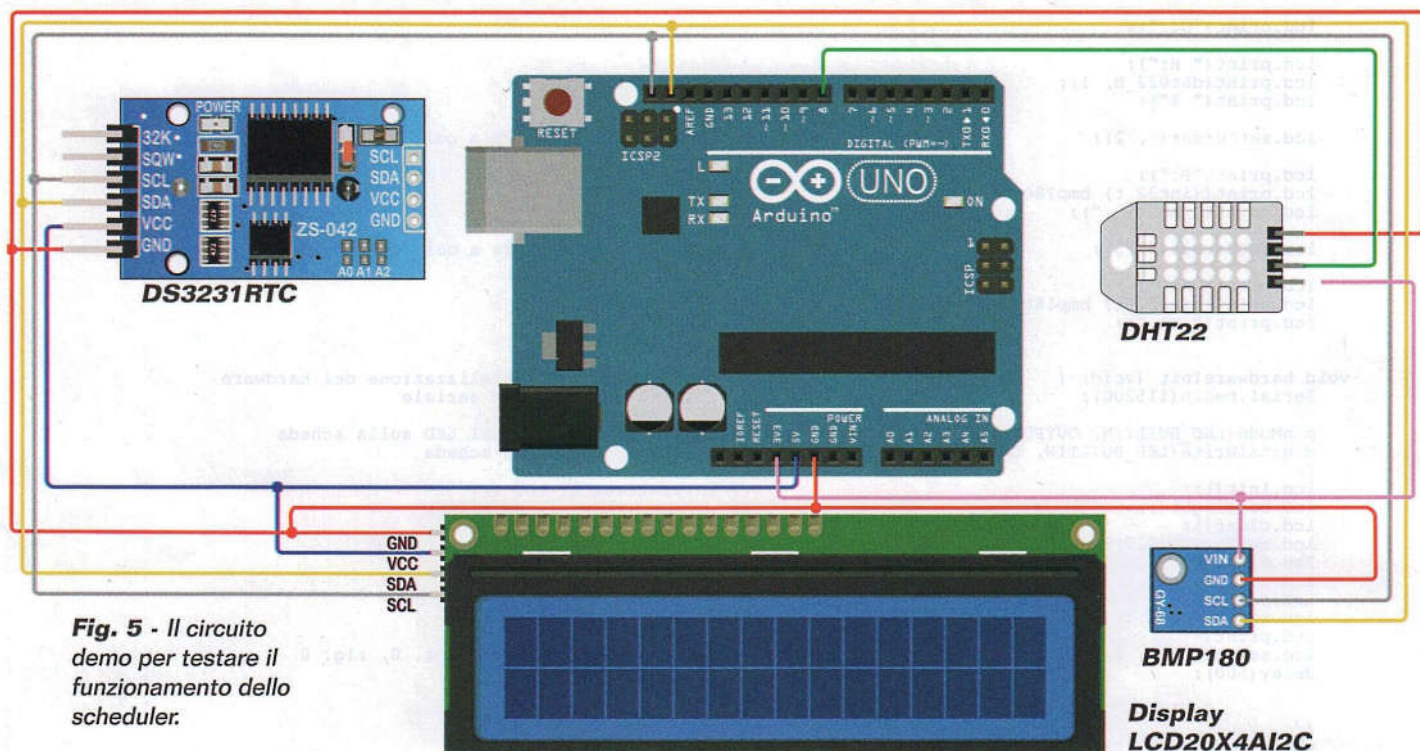



Fig. 5 - Il circuito demo per testare il funzionamento dello scheduler.

così da informare il compilatore che quello nel mezzo è puro codice 'C'.

DALLA TEORIA ALLA PRATICA

Passiamo ora a vedere un esempio pratico di utilizzo dello scheduler che riguarda un sistema dove bisogna gestire alcuni sensori, un RTC e provvedere alla visualizzazione dei dati; più esattamente il nostro sistema demo è basato su una scheda Arduino UNO rev3 che si interfaccia a un sensore di pressione BMP180 atmosferica (è un sensore barometrico utile anche a determinare l'altitudine), un sensore di temperatura e umidità DHT22, a un modulo RTC basato sul DS3231 e ad un display LCD da 4 righe x 20 colonne con interfaccia I2C-Bus. Il sensore BMP180 e l'RTC sono montati ognuno su una breakout board dedicata. Il circuito va cablatto secondo lo schema proposto nella Fig. 5 ed è illustrato dal vero, completo di tutte le parti, nella Fig. 6.

Resta inteso che il programma può essere facilmente adattato ad altri tipi di sensore.

Bene, passiamo adesso alla descrizione del firmware, nel quale abbiamo deciso di suddividere i compiti in quattro task principali: un primo task che ha il compito di leggere il sensore BMP180 e salvarne i dati, un secondo che ha il compito di leggere il sensore DHT22 e salvarne i dati, un terzo task che deve leggere i registri del RTC (DS3231) per recuperare i valori di data ed

ora e salvarli in un array e un quarto ed ultimo task che ha il compito di visualizzare i dati letti e salvati dagli altri tre task. Questi task sono chiamati ciascuno con delle tempistiche diverse: la lettura del BMP180 avviene ogni 30 secondi, quella del DHT22 ogni 60 secondi, quella del RTC ogni 0,5 secondi e la visualizzazione dei dati su LCD ogni secondo.

È stata poi prevista una funzione in cui si effettua tutta l'inizializzazione dell'hardware utilizzato e che viene chiamata dal `setup()` nella sua fase iniziale. Tutto il codice è ampiamente commentato e facilmente eseguibile; gli unici due punti da sottolineare sono:

1. in fase di inizializzazione dello scheduler ed in fase di aggiunta dei task, se viene rilevato un errore, viene inviato un messaggio sul monitor seriale ed il programma viene bloccato in un costrutto così fatto:

```
for ( ; ; ) ;
```

che, di fatto, blocca l'esecuzione del programma.

2. la funzione `loop()`, pur dovendo esistere (è richiesta dal "core di Arduino" e non può essere omessa...) è completamente vuota, perché non verrà mai raggiunta; infatti, una volta aggiunti i task allo scheduler ed avviato quest'ultimo, sarà esso a chiamare, nei tempi prestabiliti le varie funzioni, senza mai terminare.

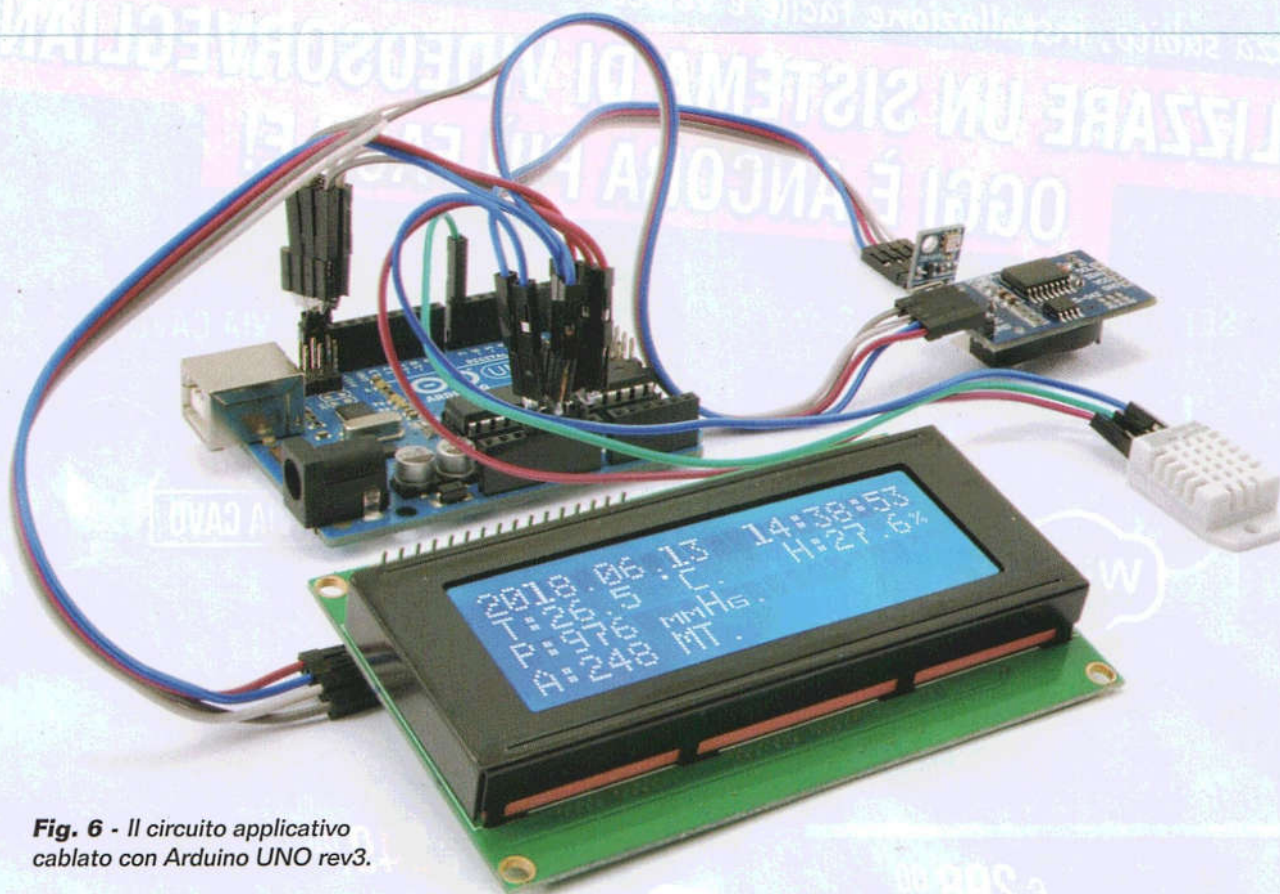


Fig. 6 - Il circuito applicativo cablatto con Arduino UNO rev3.

Con questo crediamo di avervi spiegato tutto quel che occorre a mettere in funzione lo scheduler.

CONCLUSIONI

Come accennato in apertura, quello qui descritto è uno "scheduler cooperativo" in versione estremamente semplificata, che mette a disposizione un minimo di funzioni per il corretto funzionamento ed offre un parziale controllo dei task.

Chi ne ha le capacità e la necessità, può prenderlo come base di partenza e sbizzarrirsi ad espanderlo con ulteriori funzionalità; chi invece vuole eliminarsi l'incombenza della gestione di un "super loop()", in cui è a suo carico, tramite il corretto utilizzo della funzione `millis()`, tutto il controllo delle varie attività con le tempistiche richieste, può usarlo così come è, rendendo molto più semplice e leggibile il programma. Naturalmente, come anche spiegato nella parte introduttiva, uno "scheduler cooperativo" di questo tipo ha dei limiti e sicuramente non è adatto ad applicazioni in cui sia necessario dare una priorità ai vari task, fare in modo che un task possa interromperne un altro (che poi viene ripreso nel punto in cui era stato interrotto), permette la gestione di semafori, mutex, ecc.

Per fare queste cose è necessario un vero piccolo sistema operativo, magari capace di operare in "tempo reale", in cui le varie attività hanno una durata

stabilita a priori e che vengano attivate esattamente nel momento richiesto in modo deterministico. In un prossimo articolo affronteremo un sistema operativo del genere ed esamineremo alcune delle caratteristiche peculiari di uno dei più diffusi RTOS, FreeRTOS™, in grado di essere utilizzato anche su sistemi Arduino. ■



per il MATERIALE

I componenti utilizzati per realizzare questo progetto sono disponibili presso Futura Elettronica. La board Arduino UNO REV3 (cod. ARDUINOUNOREV3) è in vendita a Euro 24,50, la breakout con DS3231 ed EEPROM (cod. DS3231RTC) è disponibile a Euro 5,90, il sensore di temperatura e umidità DHT22 (cod. DHT22) costa Euro 9,00, il sensore di pressione (cod. BMP180) è disponibile a Euro 5,50 e il display LCD 20x4 con interfaccia I²C (cod. LCD20X4AI2C) è in vendita a Euro 18,00. I prezzi sono IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287 - www.futurashop.it

Sicurezza subito, installazione facile e veloce!

REALIZZARE UN SISTEMA DI VIDEOSORVEGLIANZA OGGI È ANCORA PIÙ FACILE!

SET COMPLETO WIRELESS



Cod. **FR680**

€ 298,00

Il set è composto da un NVR a 4 canali, quattro telecamere IP IR wireless, un mouse USB, cinque alimentatori 12 Vdc, quattro antenne per le telecamere con attacco SMA, due antenne per l'NVR con attacco SMA, un cavo di rete da 1 metro, manuale utente/CD.

App
gratuite
per Android
e iOS



SET COMPLETO VIA CAVO



€ 249,00

Cod. **FR700**

Il set comprende un DVR ibrido a 4 ingressi (AHD, CVI, TVI, IP e Analogico) completo di mouse, un Hard Disk da 1000 GB, due telecamere Bullet da esterno da 2 Megapixel con illuminatori IR, due telecamere Dome Day&Night da 2 Megapixel, quattro alimentatori per le telecamere, quattro cavi video da 18 metri.

Rendi più sicura la tua casa, il tuo negozio e il tuo ufficio con questi sistemi completi di videosorveglianza e registrazione che si installano in pochi minuti e permettono la visione remota tramite PC e smartphone connessi a internet.

Prezzo IVA inclusa.

**FUTURA
ELETTRONICA**

Futura Group srl • Via Adige, 11
21013 Gallarate (VA)
Tel. 0331/79977

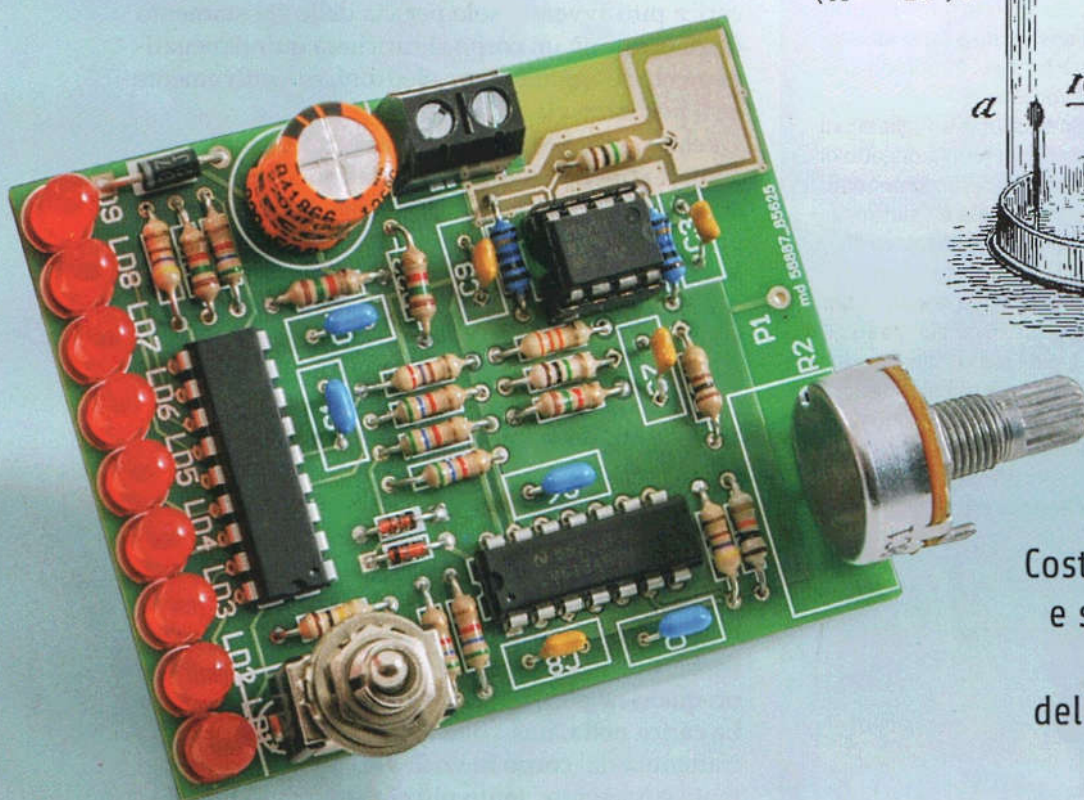
Caratteristiche tecniche e vendita on-line su:

www.futurashop.it

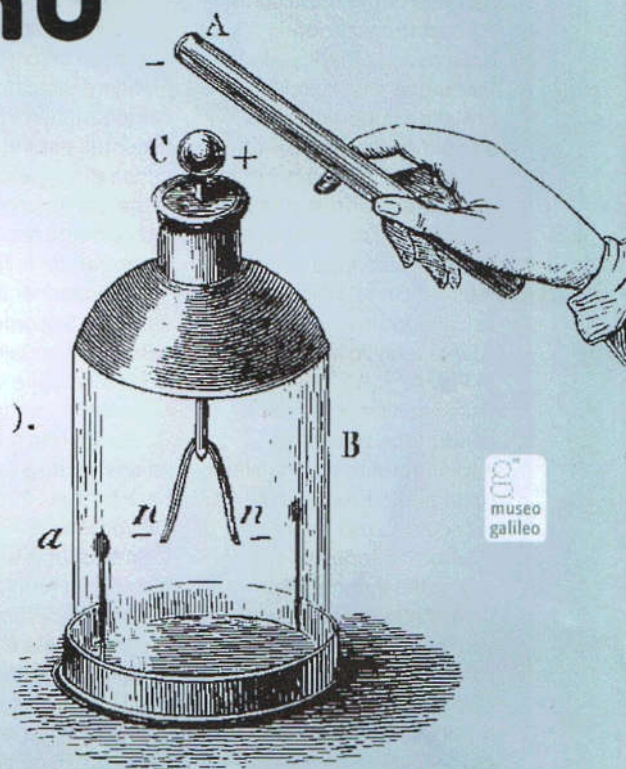


ELETTROMETRO CON BARRA DI LED

dell'Ing. OSVALDO SANDOLETTI



($h = 21$).



museo
galileo

Costruiamo un semplice
e sensibile dispositivo
per la misurazione
dell'elettricità statica...
e non solo.

Uno degli strumenti che popolano i laboratori di Fisica delle scuole e che molti di voi avranno sicuramente visto negli anni da alunni delle Superiori, è l'elettrometro, nella sua forma canonica: si tratta di un apparato anche abbastanza semplice che mostra se un corpo possiede elettricità statica. In queste pagine ve ne proponiamo una versione elettronica, modernizzata, dove l'elettricità statica posseduta da un corpo elettrizzato viene mostrata da una fila di LED.

L'ELETTRICITÀ STATICA: CAUSE ED EFFETTI

Tutti i materiali isolanti sono potenziali *contenitori* di elettricità statica, ovvero, possono **accumulare** e **trattenere** un carica elettrica positiva oppure negativa rispetto allo stato neutro, che è quello dove la quantità di carica positiva eguaglia quello della carica negativa. La carica elettrica nasce nell'atomo, che è l'elemento basilare costituente la materia, dove troviamo un nucleo composto da cariche positive, chiamate protoni, aggregate da particelle

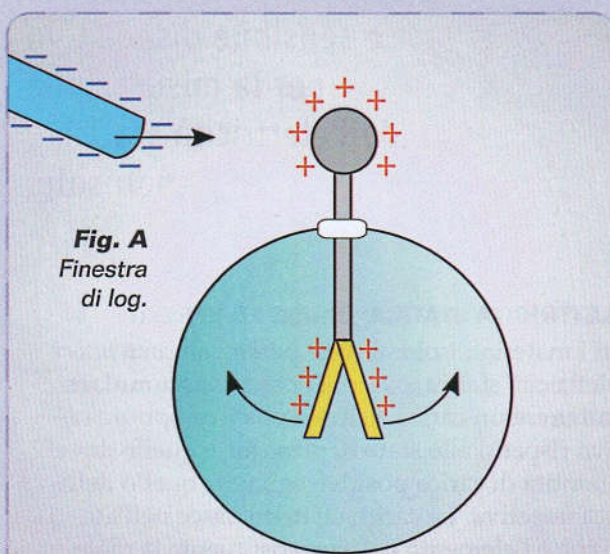
L'elettroscopio e l'elettrometro

Inventato nel diciottesimo secolo, l'elettroscopio è stato uno strumento fondamentale per la rilevazione e la caratterizzazione della carica elettrica, un fenomeno che si conosceva ormai da tempo e che si sapeva anche ricreare (era già stata inventata la prima macchina elettrostatica). Il modello più conosciuto (e tuttora utilizzato a scopo dimostrativo nei laboratori di fisica didattica) è l'elettroscopio a foglioline, illustrato in **Fig. A**.

Esso consiste in un'asta conduttrice collegata superiormente ad una sfera (o un piattello) di metallo, mentre sull'estremità inferiore vi sono appese due sottili lamine d'oro (le foglioline); queste non vengono vincolate all'estremità inferiore, rimanendo quindi libere di muoversi tra di loro: per tale motivo esse vengono poi racchiuse in un'ampolla di vetro per evitare che siano perturbate dall'aria dell'ambiente circostante. In assenza di carica netta, tra le fogliole non esiste alcuna forza at-

trattiva o repulsiva ed esse si dispongono verticalmente per via della forza di gravità, vicine e parallele tra di loro. Basta però avvicinare alla sfera un corpo carico (in figura caricato negativamente), perché questa accumuli per induzione una carica di segno opposto che viene poi trasmessa tramite la bacchetta conduttrice ad entrambe le fogliole, che vengono quindi a loro volta caricate. Siccome la loro carica è di uguale polarità, tra le due fogliole viene ad instaurarsi (per la legge di Coulomb) una forza repulsiva che tende a farle allontanare tra loro, formando una "V" rovesciata.

L'elettroscopio a fogliole d'oro è stato nel tempo oggetto di successivi perfezionamenti (degno di nota è l'elettroscopio a condensatore di Volta), ed è stato dotato anche di scale graduate per la misurazione del grado di carica (proporzionale all'angolo formato dalle foglioline d'oro): in questo contesto, la designazione più corretta dello strumento è elettrometro.



dette neutroni; le cariche negative sono rappresentate dagli elettroni, che girano intorno al nucleo in specifiche orbite (orbitali).

Malgrado abbiano massa notevolmente diversa, protoni ed elettroni possiedono uguale carica, ma di segno opposto e l'atomo è in condizione di equilibrio quando sono in uguale numero; la somma delle singole cariche fatta considerando la polarità prende il nome di *carica netta* ed è nulla solamente in particolari condizioni, in quanto ciascun atomo in condizioni normali presenta una valenza, ovvero uno scompenso di carica negli orbitali periferici. Quando si verifica uno scompenso di carica negli atomi costituenti un materiale, si dice che il corpo è *carico*. Dato che i protoni sono interni al nucleo dell'atomo e non possono dunque muoversi, l'accumulo di carica può avvenire solo per via dello spostamento degli elettroni: un corpo si caricherà quindi negativamente se aggiungiamo elettroni, e positivamente se li portiamo via.

Questo accumulo si può ottenere in tre modi: per contatto (mettendo un corpo già carico in contatto con un altro vi è una distribuzione delle cariche in eccesso tra il primo ed il secondo), per induzione (mettendo un corpo già carico vicino ad un altro, questi si caricherebbe con segno opposto rispetto al primo) oppure per strofinio (sfregando tra di loro due corpi di materiale diverso, anche se inizialmente tutti e due fossero nello stato neutro).

Quest'ultimo fenomeno (detto effetto *triboelettrico*, dal greco *tribos* = sfregamento) è il maggiore responsabile del caricamento elettrico di quasi tutti gli oggetti isolanti con cui abbiamo a che fare nel nostro quotidiano: è infatti sufficiente camminare perché le nostre scarpe (se hanno una suola isolante) sfreghino sul terreno e accumulino cariche elettriche; oppure, sfregando un panno sintetico contro un qualsiasi altro oggetto isolante, e così via.

La carica netta, una volta accumulata, viene poi trattenuta dal corpo in virtù del suo isolamento: tanto è maggiore, tanto più ci metteranno le cariche a ridistribuirsi e/o "scaricarsi", verso la terra o altri oggetti aventi carica diversa. Anche l'ambiente circostante gioca un ruolo importante: climi molto asciutti favoriscono il trattenimento delle cariche, mentre l'umidità riduce l'isolamento accelerando la scarica del corpo.

Ma come facciamo a sapere se un corpo è "carico"? Per osservarlo a livello macroscopico è necessario dargli modo di scaricarsi rapidamente (ad esempio, collegandolo a terra attraverso un conduttore), in quanto il repentino flusso di cariche si traduce in un elevato, seppur istantaneo, picco di corrente elettrici-



Fig. 1 - Schema a blocchi dell'elettrometro qui proposto.

ca che si manifesta spesso (se la carica accumulata è sufficientemente elevata) come un minuscolo arco elettrico accompagnato dal classico crepitio: è la cosiddetta *scarica elettrostatica*. Spesso capita anche a noi, seppur involontariamente, di fare da conduttore elettrico e così, toccando oggetti carichi (oppure siamo noi a esserlo per via delle scarpe con suola isolante), avvertiamo anche la tipica scossa.

Un sistema di misurazione più rigoroso non deve invece basarsi sul processo di scarica del corpo, ma avere una sensibilità sufficiente a rilevare le cariche (per contatto o per induzione), senza che esse vengano alterate in maniera sensibile. Qui entra in gioco l'elettrometro che è fondamentalmente, nella nostra variante "elettronica", un voltmetro avente un **ingresso ad altissima impedenza** in modo da riuscire a leggere il potenziale captato da un elettrodo rilevatore della carica. Questa carica (Q) è infatti legata al potenziale V dalla relazione:

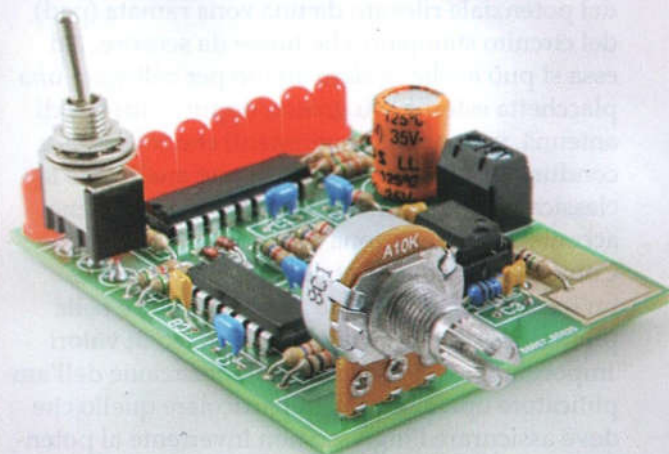
$$Q = CV$$

dove C è la capacità formata dall'oggetto portatore di carica (costituente un'armatura del nostro condensatore virtuale) e il riferimento di tensione del circuito (la terra, che fa quindi da seconda armatura). Siccome tale capacità non è nota o facilmente calcolabile (dipende tra l'altro da molti fattori tra i quali la nostra stessa vicinanza all'oggetto), non è possibile effettuare misurazioni quantitative del potenziale assoluto dell'oggetto.

In Fig. 1 possiamo vedere lo schema a blocchi generale di un tale dispositivo: l'elettrodo rilevatore di elettricità statica è costituito da una barretta di rame o altro conduttore, di superficie opportuna per

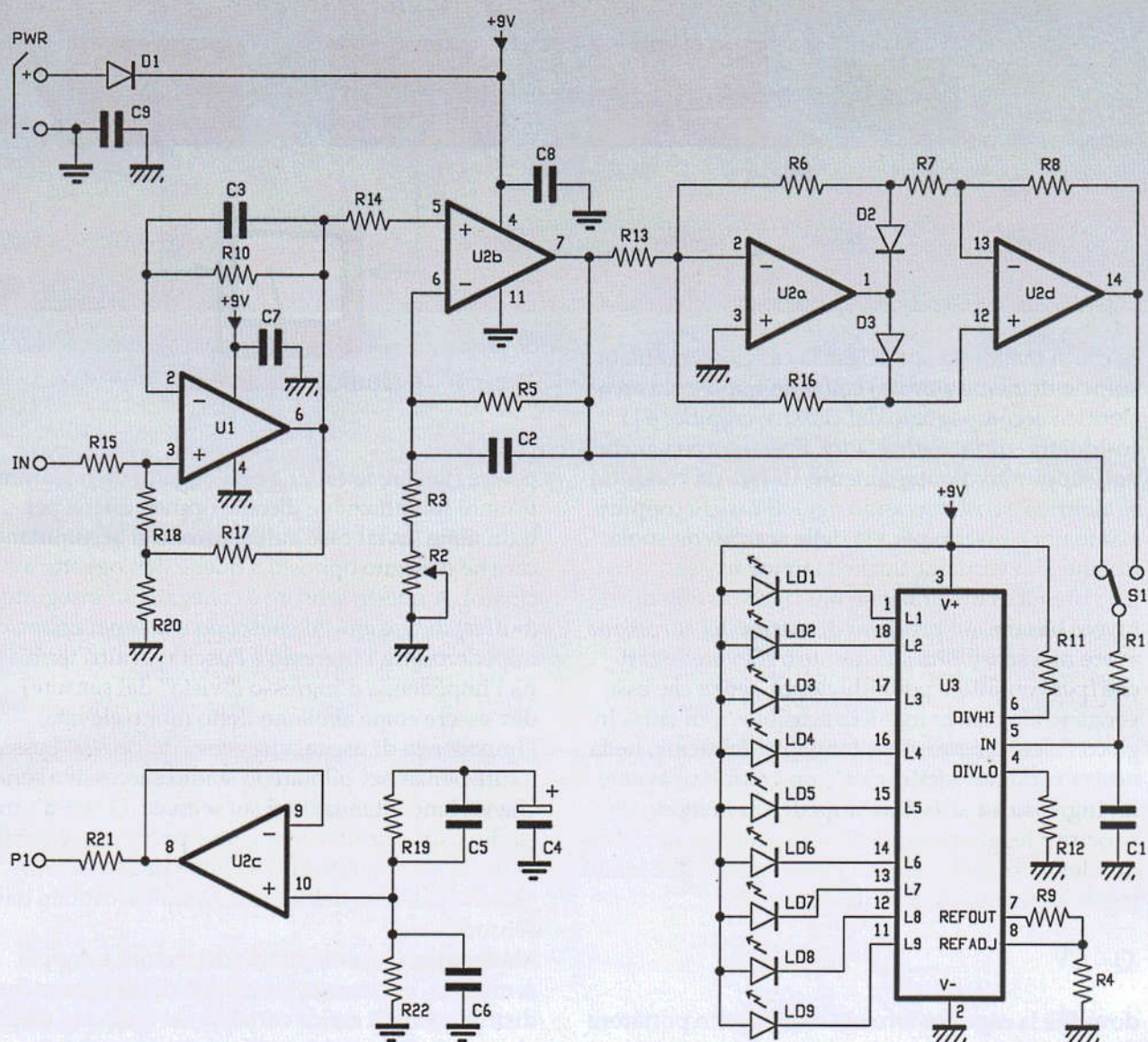
potere rilevare le cariche dell'oggetto da misurare tramite contattazione diretta, oppure anche per induzione (in tal caso sull'elettrodo si accumulano cariche di segno opposto a quelle dell'oggetto avvicinato). A questo sensore è collegato un inseguitore (buffer) di tensione, il cui scopo è la separazione di impedenza tra l'ingresso e l'uscita: in altri termini, l'impedenza d'ingresso ("vista" dal sensore) dev'essere come abbiamo detto molto elevata; l'impedenza di uscita, viceversa, dev'essere bassa a sufficienza per pilotare lo stadio successivo senza che vi siano attenuazioni sul segnale. Quest'ultimo stadio è un amplificatore vero e proprio, regolabile in modo da adattare la sensibilità della visualizzazione al livello dell'elettricità statica captata dal sensore.

Abbiamo poi inserito un raddrizzatore a doppia semionda, per dare la possibilità di rilevare anche distribuzioni di carica variabili nel tempo (o meglio, alternate). Queste danno luogo a campi elettrici pure variabili, che possono essere captati e rilevati dal nostro strumento. Come applicazione pratica di questa caratteristica, si può rilevare la vicinanza dello strumento ai fili dell'impianto elettrico domestico, che generano un campo alternato a 50Hz.



CARATTERISTICHE TECNICHE

- Visualizzazione: barra di 9 LED (con zero centrale)
- Tensione di alimentazione: 9-12 Vcc
- Assorbimento tipico: 20 mA
- Tensione di ingresso di fondoscala minima: +/-200 mV
- Banda passante (senza raddrizzamento): 0÷2 Hz



SCHEMA ELETTRICO

Diamo ora uno sguardo allo schema elettrico del nostro elettrometro; per la descrizione partiamo da U1, utilizzato come inseguitore per la lettura del potenziale rilevato da una zona ramata (pad) del circuito stampato, che funge da sensore. Ad essa si può anche saldare un filo per collegare una placchetta esterna più grossa, oppure altri tipi di antenna, in base agli esperimenti che si desidera condurre: per esempio se vogliamo analizzare la classica bacchetta di vetro elettrizzata possiamo accontentarci della zona ramata del PCB. Come descritto prima, tale stadio deve esibire un'impedenza d'ingresso molto elevata, e nella pratica questo implicherebbe l'utilizzo di valori "importanti" dei resistori di polarizzazione dell'amplificatore operazionale, in particolare quello che deve assicurare l'ingresso non invertente al potenziale di riferimento (GNDF nello schema), in quanto

da esso dipende l'impedenza vista dal sensore. Ma avendo resistenze di valore elevato (dell'ordine delle centinaia di $M\Omega$) è necessario utilizzare componenti con bassissime correnti di polarizzazione d'ingresso (dell'ordine dei picoampere o anche meno), come è ad esempio l'operazionale LMC6001, che però sono anche molto costosi. Abbiamo pertanto scelto di ricorrere ad un artificio circuitale che consenta di utilizzare resistori di valore "moderato" (sui $10M\Omega$), e quindi anche un amplificatore operazionale di uso comune, pur senza abbassare l'impedenza d'ingresso vista dal sensore. Questo sistema è basato sull'applicazione del teorema di Miller che adesso descriveremo brevemente nel seguito, in quanto può essere adottato anche in altre applicazioni. Come schematizzato in Fig. 2, il teorema di Miller afferma che un'impedenza Z posta tra l'ingresso e l'uscita di un amplificatore di tensione (schema A) risulta equivalente ad un circuito con

due impedenze poste tra l'amplificatore e la massa (schema B) i cui valori dipendono sia dall'impedenza Z che dall'amplificazione K tra ingresso ed uscita. In particolare, se abbiamo a che fare con un amplificatore non invertente (come è nel nostro caso U1, essendo configurato come inseguitore a guadagno unitario), K è positivo e il valore dell'impedenza di ingresso Z_1 potrà in teoria variare tra Z (se $K=0$) e infinito ($K=1$).

Per ottenere valori "intermedi", nel nostro schema riduciamo il guadagno ad un valore inferiore all'unità tramite il partitore formato da R17 ed R20:

$$K = R20 / (R17 + R20)$$

Con i valori suggeriti dal nostro schema elettrico, otteniamo $K=0,968$. Da tale partitore viene prelevato, tramite la resistenza R18 (che fa le veci dell'impedenza Z di figura 3-A), il segnale di reazione da riportare sull'ingresso non invertente dell'amplificatore. Essendo $R18 = 10M\Omega$, in base al teorema di Miller è come se applicassimo tra l'ingresso di U1 e la massa un resistore da:

$$10/(1-K)=10/(1-0,968)=310 M\Omega$$

Quindi anche l'impedenza d'ingresso vista dal sensore (escludendo le correnti di polarizzazione e offset di U1, comunque ancora trascurabili rispetto a queste impedenze) è pressoché uguale.

Per i nostri scopi, questo valore si è dimostrato adeguato; è comunque chiaro che avvicinando il valore di K all'unità (modificando R17 e/o R20) si potranno ottenere valori anche ben più elevati di impedenze d'ingresso, tenendo in considerazione che

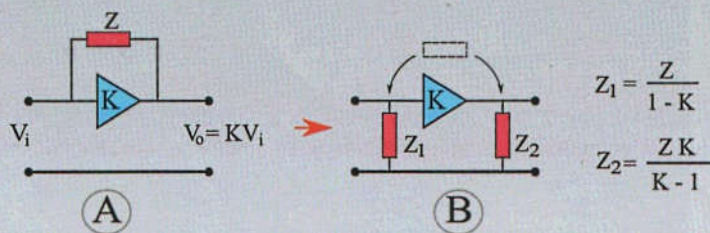


Fig. 2 - Schematizzazione del teorema di Miller.

converrebbe in tal caso sostituire comunque anche l'operazionale utilizzato (TL061) con un dispositivo che presenti correnti di ingresso inferiori. Inoltre, siccome questo metodo prevede la presenza di una reazione positiva dell'operazionale (il segnale di uscita viene in parte riportato sull'ingresso non invertente), avvicinando K a 1 aumenterà la tendenza all'instabilità.

Abbiamo così spiegato lo scopo delle resistenze R17, R18 ed R20 collegate all'operazionale U1. La resistenza R10 (che per minimizzare l'offset dell'operazionale dev'essere possibilmente di valore pari a $R18+R20$) realizza la controreazione unitaria, perché come già detto U1 deve lavorare come inseguitore. In tal modo, la tensione di uscita ricopia il valore "captato" dalla sonda ma, essendo a bassa impedenza, lo utilizziamo anche per pilotare l'"anello di guardia" disegnato sul circuito stampato, intorno e sotto alla pista del segnale d'ingresso. Attraverso R14 il segnale di U1 arriva quindi all'operazionale U2B utilizzato come amplificatore di tensione, il cui guadagno è regolabile attraverso il potenziometro R2; con i valori proposti, esso potrà essere impostato da un minimo di 2 ad un massimo di 22 volte, a seconda della sensibilità che si desidererà ottenere sulla lettura.

Il segnale così amplificato viene prelevato dall'usc-

Api robot ed elettricità statica

Walmart, nota catena commerciale americana, sta ideando una propria versione di fattoria automatizzata, in cui voleranno flotte di api artificiali. L'idea di Walmart in vero non è nuova, perché già da tempo si conducono esperimenti finalizzati a far fronte al calo della popolazione degli insetti impollinatori, causato tra le altre cose dai cambiamenti climatici e dall'abuso di pesticidi.

Le api robotiche, tecnicamente sono dei droni per l'impollinazione e dispongono di telecamere e sensori per rilevare il polline, raccoglierlo e trasportarlo da un fiore all'altro. Per far attaccare questi minuscoli droni alle superfici, si ipotizza di sfruttare l'elettricità statica, come già fatto nel progetto RoboBees realizzato dalla Harvard University.



Gli anelli di guardia

Quando si vogliono realizzare circuiti con linee ad altissima impedenza bisogna tenere in considerazione anche le correnti di fuga (o di perdita) che possono partire da queste linee e scorrere lungo oppure all'interno dei materiali utilizzati per costruire il circuito stesso, per richiudersi a massa e degradare quindi il livello del segnale utile. Questo succede perché nessun materiale è un isolante perfetto, e nei confronti di una differenza di potenziale applicata tra due punti si comporta (in prima approssimazione) come una resistenza, di valore certamente elevato, ma non infinito. Ad esempio, tra due piste superficiali adiacenti di un circuito stampato esiste una certa resistenza superficiale, dovuta al tipo di *solder-resist* che le ricopre (se presente), alla pulizia della superficie (polvere, disossidante della saldatura) e al grado di umidità

dell'aria sovrastante. Persino il materiale componente il circuito stampato stesso presenta una propria resistenza, e dunque può instaurarsi una piccola corrente "di fuga" anche tra due conduttori, se posti a differente potenziale, che scorrono sulle due facce opposte del circuito.

Queste resistenze (cosiddette *distribuite* in quanto non sono componenti veri e propri), in virtù del loro valore elevato, non vengono normalmente prese in considerazione nello studio di un circuito; tuttavia quando bisogna andare a leggere il segnale proveniente da un sensore con impedenza dell'ordine delle centinaia di Megaohm, il loro contributo diventa significativo.

Se ad esempio il nostro circuito prevedesse uno stadio amplificatore con impedenza d'ingresso di $1G\Omega$, e le resistenze parassite verso massa della

pista cui è collegato il sensore fossero simili, metà della corrente erogata da quest'ultimo fluirebbe a massa attraverso queste resistenze prima di arrivare all'amplificatore!

Le contromisure attuabili sono due: la prima consiste nell'effettuare collegamenti volanti (in aria) dei fili che portano i segnali ad alta impedenza, dal sensore e fino all'ingresso dell'amplificatore. La resistenza dell'aria, specialmente se secca, è infatti notevolmente superiore a quella superficiale e trasversale dei circuiti stampati.

Il secondo sistema non mira invece ad aumentare le resistenze, ma a neutralizzarne gli effetti. In base alla legge di Ohm, la corrente che scorre attraverso una resistenza è proporzionale alla differenza dei potenziali ai suoi estremi; pertanto, se questa differenza si annullasse, la corrente pure

si annullerebbe indipendentemente dal valore della resistenza, reale o distribuita che sia. La conclusione è che il metodo per annullare le correnti parassite che possono partire da un conduttore è quello di circondarlo con una sorta di "barriera" (detta *guardia*) posta allo stesso suo potenziale. Come esempio si consideri la **Fig. B**, sulla cui sinistra è illustrata la parte di un circuito stampato su cui scorre la pista di segnale (ad alta impedenza) S, che si trova al potenziale di 1V, ed una vicina pista di massa M. Per via delle (inevitabili) resistenze superficiali qui raffigurate dal resistore R_s , si instaurerà una corrente di perdita I_p dalla pista di segnale verso la massa.

Le cose cambiano se ci spostiamo nella figura di destra, nella quale è stata interposta tra le piste S e M una pista di guardia G allo stesso potenziale di S.

ta di U2B ed inviato al circuito composto da U2A e U2D, che realizzano il raddrizzatore ideale a doppia semionda. Questo stadio ci permetterà, come già anticipato, di effettuare la rilevazione di campi elettrici alternati perché il segnale raddrizzato presente sull'uscita di U2D potrà "superare" il successivo filtro passa-basso formato da R1 e C1.

Il deviatore S1 permette comunque di scegliere se visualizzare il segnale raddrizzato oppure no: dal suo contatto comune il segnale selezionato viene infatti inviato, tramite il succitato filtro R1-C1, al visualizzatore a LED. Per quanto riguarda quest'ultimo stadio si è fatto ricorso all'integrato LM3914 (U3), in quanto contiene tutta la circuiteria necessaria per realizzare un voltmetro a 10 led; per una descrizione più approfondita del suo funzionamento rimandiamo all'articolo di "Rilevatore di microspie" presentato sul numero 219 di *Elettronica In*. In questa applicazione lo utilizziamo come voltmetro a 9 led con visualizzazione a singolo punto; le resistenze R11 e R12 sono state inoltre calcolate in modo che quando il suo ingresso (piedino 5) si trova sulla metà della tensione di alimentazione, sia acceso il led centrale: a tale tensione si trova infatti il potenziale della massa di segnale GNDF, che corrisponde all'assenza di segnali in ingresso all'elettrometro. In tal modo, quando in ingresso applicheremo un segnale negativo, il punto luminoso si sposterà

verso sinistra, e viceversa con un ingresso positivo. Se invece tramite S1 selezioneremo il segnale proveniente dal raddrizzatore (uscita di U2D), la visualizzazione sarà sempre positiva (il punto luminoso potrà trovarsi nella sola metà destra della barra). Infine U2C viene utilizzato, insieme al partitore R19-R22, per generare il potenziale di massa virtuale (GNDF) del sistema, posizionato a metà della tensione di batteria, che serve al funzionamento degli stadi di amplificazione e di raddrizzamento.

MONTAGGIO E COLLAUDO

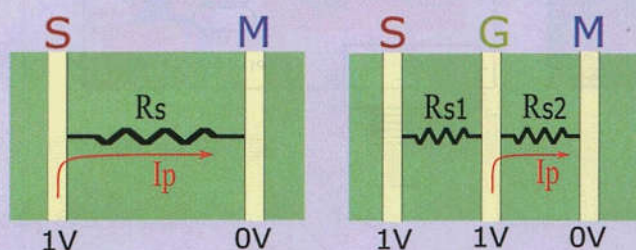
Il montaggio del circuito non presenta particolari difficoltà, in quanto tutti i componenti sono del tipo "tradizionale" (a foro passante): bisogna solo prestare attenzione al verso dei componenti polarizzati (diodi, condensatori elettrolitici e circuiti integrati) indicato nel piano di montaggio.

Per la saldatura, è consigliabile l'utilizzo di una lega a basso contenuto di fluossido soprattutto nella zona dell'amplificatore ad alta impedenza, in quanto i residui della saldatura concorrono all'abbassamento dell'impedenza d'ingresso, anche se tale effetto viene mitigato dalla presenza degli anelli di guardia disegnati sul circuito stampato. Ultimato il montaggio, si può fornire tensione al circuito; per la prima accensione è consigliabile utilizzare un alimentatore stabilizzato da 9V e limitato

Tra le tre piste sarà sempre presente una resistenza di perdita superficiale (rispettivamente R_{s1} e R_{s2}); tuttavia, essendo nulla la differenza tra i potenziali di S e G, su R_{s1} non vi sarà alcuna corrente di perdita. Questa corrente sarà invece presente su R_{s2} , in quanto tra G e M insiste la differenza di potenziale di 1V; però questa corrente viene stavolta fornita dal circuito che pilota la linea di guardia G, senza più "caricare" la linea di segnale S. Questo ragionamento suggerisce che la linea di guardia debba essere pilotata da un generatore di tensione con le seguenti due caratteristiche:

- 1) La sua tensione deve copiare quella presente sulla linea da proteggere (per non avere mai differenze di potenziale su R_{s1});
- 2) deve avere un'impedenza sufficientemente bassa da fornire tutte le possibili correnti di perdita senza che la sua tensione venga alterata (per soddisfare sempre e comunque il punto 1).

Fig. B



Come criterio di tracciamento della pista di guardia sul circuito stampato si è soliti, per prevenire correnti di fuga in qualsiasi direzione, circondare completamente le linee di segnale da proteggere: da qui il termine anelli di guardia (guard ring).

Inoltre, per prevenire anche le correnti di fuga trasversali del circuito, sarebbe buona norma disegnare un piano di rame sulla faccia opposta a quella su cui scorrono le piste da proteggere; questo piano dev'essere poi collegato tramite fori passanti (vias) all'anello di guardia sovrastante (quindi è sufficiente che il piano ricalchi il disegno dell'anello di guardia).

Questo si può fare a patto che il circuito sia almeno a doppia faccia, e non vi siano vincoli di impedenza della linea di segnale da rispettare. In applicazioni particolari, per limitare

le correnti trasversali e le cosiddette correnti di rilassamento del dielettrico, si preferisce addirittura non adottare come materiale del circuito stampato il comune FR4 ma orientarsi su particolari laminati ceramici quali il Roger4350 (frequentemente utilizzato nei circuiti a microonde).

Il metodo del conduttore di guardia non vale solo per i circuiti stampati, ma si applica anche nei collegamenti "volanti": allo scopo esistono infatti cavetti coassiali detti "triassiali", formati da un conduttore centrale e due schermi concentrici (uno più interno e l'altro più esterno) isolati tra loro. Se il segnale ad alta impedenza passa nel conduttore centrale, si può applicare la tensione di guardia sullo schermo interno, mentre il ritorno di segnale (ovvero la massa) viene applicato allo schermo esterno.

(se possibile) in corrente: l'assorbimento del circuito dev'essere intorno ai 20mA e dovrebbe illuminarsi il LED centrale della barra.

A questo punto, è sufficiente sfregare il capo di una biro su un panno e, tenendola dalla parte opposta, avvicinarla alla zona ramata che funge da sensore, per vedere uno spostamento del punto luminoso; allontanando poi l'oggetto, il punto luminoso dovrà spostarsi dalla parte opposta. Ciò accade perché avvicinando l'oggetto al sensore aumenta la capacità tra i due oggetti, e viene indotto sul sensore un potenziale con un certo segno; allontanando l'oggetto, la variazione di capacità diventa negativa, inducendo così sul sensore un potenziale di segno opposto. L'effetto sarà tanto più marcato quanto più velocemente si avvicinerà e si allontanerà la biro dal sensore. Se il punto luminoso non dovesse muoversi, si potrà giocare sulla sensibilità del dispositivo agendo sul potenziometro R2.

Un'altra verifica del funzionamento si può effettuare commutando il deviatore S1 per prelevare il segnale da visualizzare dall'uscita di U2D, ovvero inserendo il raddrizzatore a doppia semionda. In tal modo, avvicinando il circuito ad una linea dell'impianto elettrico a 230V (per esempio, vicino ad una presa di corrente) si potrà osservare lo spostamento del punto luminoso verso destra: in questo caso il sensore non sta captando un campo elettrico dov-

uto a cariche statiche, ma quello alternato prodotto dalla tensione di rete.

Raccomandiamo di non tenere il circuito in mano, o quantomeno non toccarne le piste oppure il sensore, in quanto il nostro corpo è un'ottima "antenna" del campo elettrico della rete, nonché sede di cariche statiche (specialmente se calziamo scarpe con suola isolante): consigliamo quindi di racchiuderlo in un piccolo contenitore plastico, lasciando esposta solo la zona del sensore.

MISURAZIONE DELL'IMPEDEZZA D'INGRESSO

Volendo, si potrebbe verificare quale sia l'effettiva impedenza d'ingresso del nostro elettrometro (almeno nella sua componente resistiva): allo scopo si può utilizzare lo schema suggerito in Fig. 3, che richiede l'impiego di un piccolo alimentatore (o di

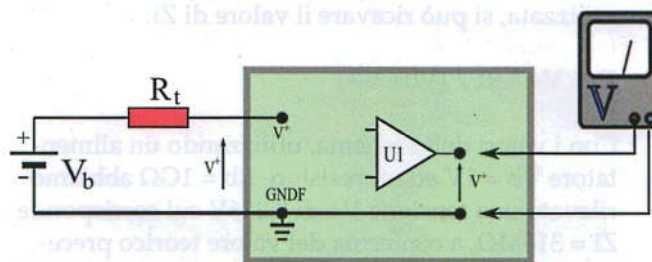
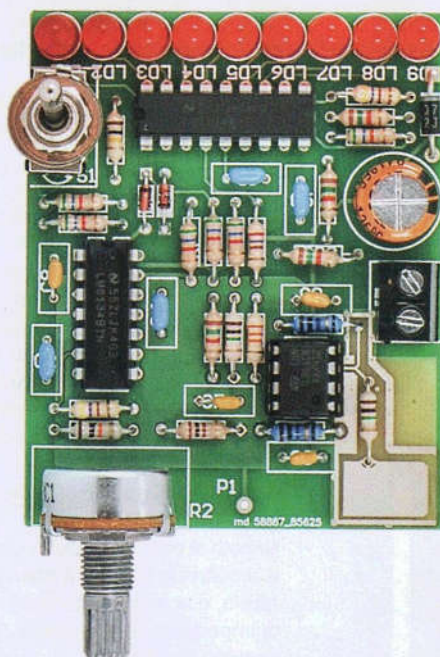


Fig. 3 - Schema di prova impedenza.

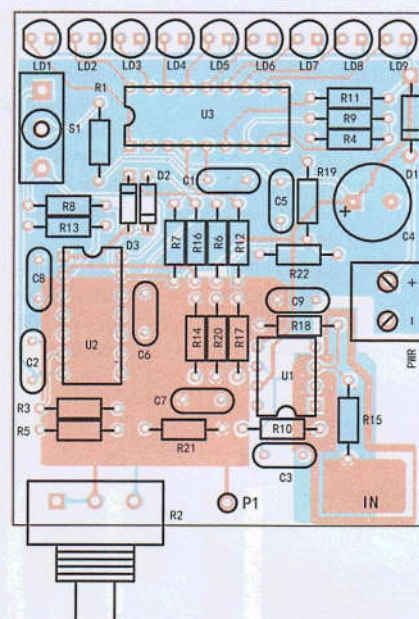
[piano di MONTAGGIO]

Elenco Componenti:

R1: 100 kohm
 R2: potenziometro 10 kohm
 R3, R11: 470 ohm
 R4, R6÷R8: 5,6 kohm
 R5: 10 kohm
 R9, R14, R19, R22: 1,5 kohm
 R10: 11 Mohm
 R12: 270 ohm
 R13, R16: 5,6 kohm
 R15, R20: 1 Mohm
 R17: 33 kohm
 R18: 10 Mohm
 R21: 100 ohm
 C1, C2: 1 µF ceramico
 C3: 1 nF ceramico
 C4: 220 µF 25 VL elettrolitico
 C5, C6: 1 µF ceramico
 C7÷C9: 10 nF ceramico
 LD1÷LD9: LED 5 mm rosso
 D1: 1N4007
 D2, D3: 1N4148
 U1: TL061CN
 U2: LM6134BIN
 U3: LM3914N
 S1: Deviatore a levetta da CS



Varie:
 - Zoccolo 4+4
 - Zoccolo 7+7



- Zoccolo 9+9
 - Morsetto 2 vie
 - Circuito stampato S1415 (67x54 mm)

una batteria) Vb in grado di fornire una tensione qualsiasi tra 1 e 2V, e di un resistore Rt di valore elevato (dell'ordine delle centinaia di MΩ, comparabilmente con il valore di impedenza d'ingresso che ci aspettiamo). Collegato il circuito all'elettrometro (che dev'essere naturalmente acceso, tramite batteria oppure un altro alimentatore), si può andare quindi a leggere la tensione sull'uscita di U1 il quale, essendo configurato come inseguitore, dovrà riproporre (rispetto a GNDF) lo stesso valore presente sull'ingresso non invertente V+, che è dato dal partitore tra Rt e l'impedenza d'ingresso Zi

$$V+ = Vb * Zi / (Zi + Rt)$$

Quindi, conoscendo qual è la Vb applicata e la Rt utilizzata, si può ricavare il valore di Zi:

$$Zi = V+ * Rt / (Vb - V+)$$

Con i valori dello schema, utilizzando un alimentatore Vb = 1V ed un resistore Rb = 1GΩ abbiamo rilevato una tensione V+ di 0,236V cui corrisponde Zi = 310MΩ, a conferma del valore teorico precedentemente calcolato. Ribadiamo che la tensione V+ dev'essere letta sull'uscita di U1 e non sull'ingresso

dell'elettrometro (a valle di Rt) in quanto l'impedenza del multimetro, seppur elevata, andrebbe ad alterare la misura.

Ricordate, infine, che le misure di tensione vanno effettuate utilizzando come riferimento la massa del circuito GNDF e non il negativo di alimentazione.

CONCLUSIONI

Lo strumento qui presentato ci permette di assumere maggiore consapevolezza sull'esistenza delle cariche elettriche che ci circondano nella vita comune, cosa che normalmente non consideriamo ma che possono essere anche fonte di problemi, ad esempio quando maneggiamo i sensibilissimi componenti elettronici odierni senza le dovute precauzioni. Al di là della valenza didattica, la metodologia di rilevazione dell'elettrometro (che è di fatto un sensore di campo elettrico) trova diverse applicazioni pratiche, tra cui la misurazione dell'inquinamento elettromagnetico a bassa frequenza, la captazione del campo elettrico prodotto dalle scariche temporalesche e persino la rilevazione "a distanza" di segnali biomedici quali l'elettrocardiogramma (chi volesse approfondire questi argomenti potrà trovare parecchia documentazione digitando "electric potential sensor" in un motore di ricerca).

ARRIVA IL SALVA BEBÈ GSM!



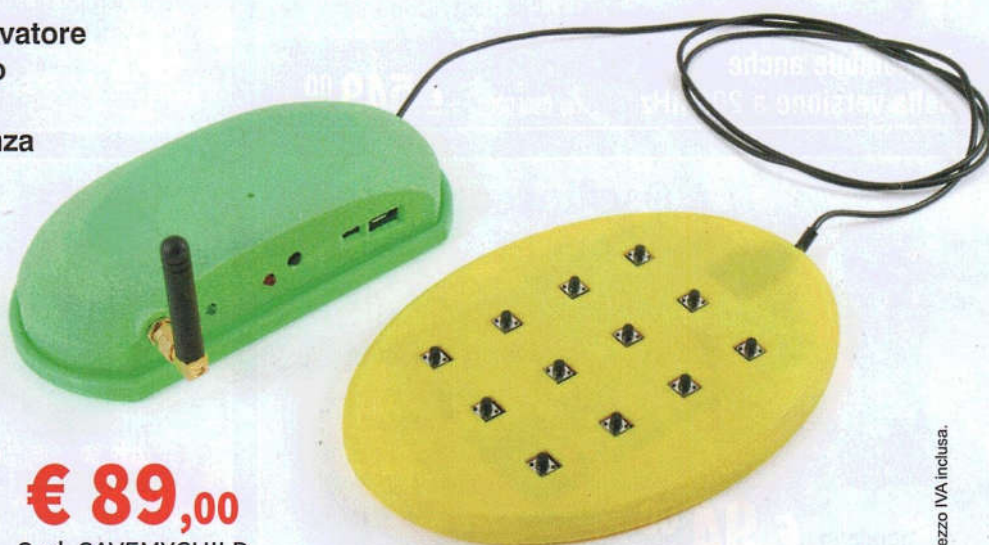
SAVE MY CHILD è un set studiato per la serenità dei genitori e per la sicurezza dei bambini. Installato in auto, grazie a un rilevatore posto sulla seduta del seggiolino ci avverte tramite un SMS e una telefonata se ci allontaniamo senza portare con noi il bambino.

LA CONFEZIONE COMPRENDE:

la scheda base con modulo GSM SIM800, l'antenna GSM, il sensore presenza bebè, l'accelerometro triassiale e cavi di collegamento tra scheda base e sensore presenza.

La batteria al litio da 3,7 V / 500 mA (necessaria nel momento in cui l'alimentazione principale viene a mancare) e il contenitore plastico (disponibile in diversi colori) sono acquistabili separatamente.

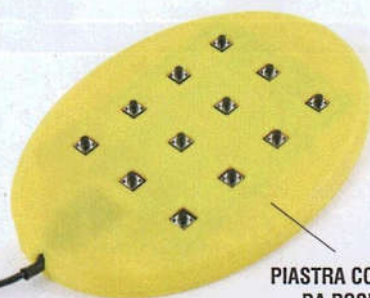
SAVE MY CHILD Set completo



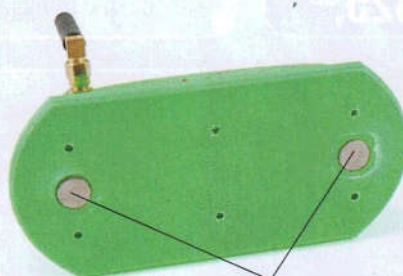
€ 89,00

Cod. SAVEMYCHILD

Prezzo IVA inclusa.



PIASTRA CON PULSANTI
DA POSIZIONARE
SOTTO IL RIVESTIMENTO
DEL SEGGIOLINO



BASE MAGNETICA

ALLOGGIAMENTO
PER BATTERIA
SUPPLEMENTARE

MODULO GSM
SIM800



N.B. La scheda base e il sensore presenza vengono forniti già montati. È solo necessario effettuare il cablaggio tra la scheda base e il sensore presenza.

VASTA GAMMA DI STRUMENTI PER IL LABORATORIO

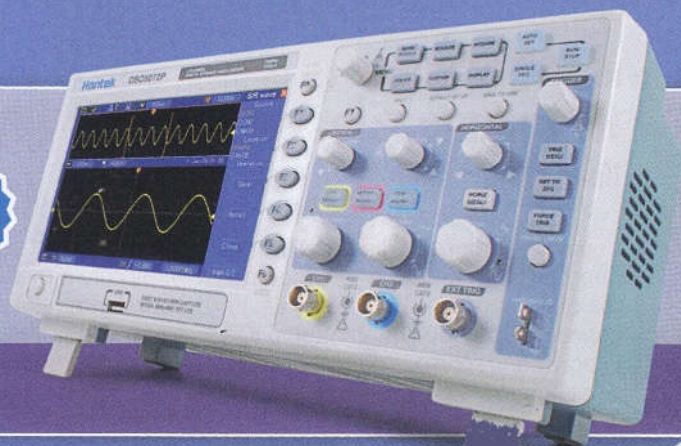
Oscilloscopio 2 Canali 70 MHz

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

COD. DS072

€ 325,00

IL PIÙ
VENDUTO



Disponibile anche
nella versione a 200MHz

COD. DS0202

€ 519,00

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

COD. DSOFG102

€ 420,00

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

Disponibile anche
nella versione a 200MHz

COD. DSOFG202

€ 549,00

Oscilloscopio a forma di penna - 1 canale

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

COD. PS02020

€ 95,00



Oscilloscopio USB per PC 2 Canali 20 MHz



COD. DSOPC22

€ 94,00

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

Disponibile anche
nella versione a 100MHz

COD. DSOPC102

€ 325,00

Generatore di funzioni e forme d'onda arbitrarie

USCITA FORME D'ONDA ARBITRARIE

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

FREQUENZIMETRO

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

€ 224,00

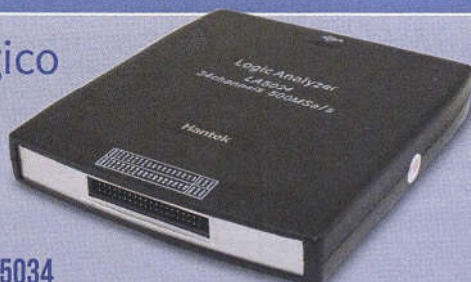
COD. FAW1025G



Analizzatore logico con 34 Canali 500 Msa/S

€ 154,00

COD. ANALOG5034



- ✓ 34 Canali
- ✓ Impedenza d'ingresso 200 Kohm
- ✓ Tensione d'ingresso da -60 V a 60 V
- ✓ Frequenza di campionamento max. 500 MHz
- ✓ Larghezza di banda segnale d'ingresso max. 150 MHz
- ✓ Ingresso compatibile TTL, LVTTTL, CMOS, LVCOMS, ECL, PECL, EIA
- ✓ Interfaccia porta USB

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

BITCOIN & BLOCKCHAIN

Nella società di oggi tutto evolve rapidamente: lo stile di vita, le comunicazioni... perfino il concetto di lavoro. Ma ci sono cose che non cambiano, come i presupposti dell'economia, sui quali tutti noi basiamo la vita di tutti i giorni: per acquisire un bene serve qualcosa che consenta lo scambio e che per elezione è il denaro. Oggi però, l'evoluzione tecnologica vuole provare a cambiare radicalmente il nostro concetto di scambio e di denaro. In questo articolo approfitteremo della grande notorietà

assunta dal Bitcoin per analizzare il sistema che ne è alla base e che trova applicazione non solo nel Bitcoin: la blockchain. Vedremo come funziona e a che cosa serve. Il fenomeno è, ormai, abbastanza diffuso e molte persone si stanno avvicinando al mondo della blockchain, del Bitcoin e delle criptovalute più in generale, tuttavia troppo spesso non hanno competenze dal punto di vista tecnologico, matematico né economico per farlo scientemente. Proveremo, quindi, a rispondere ad alcune delle domande più comuni,

di PIERO BOCCADORO

Approfittiamo di uno dei più noti esempi della vita reale per introdurre il concetto di blockchain, le sue caratteristiche e l'utilità in pratica, spiegando anche come minare i bitcoin utilizzando una piattaforma hardware essenziale come Raspberry Pi.



Fig. 1 - Rappresentazione del Bitcoin.

su cosa sia e cosa voglia dire la blockchain, come funziona il Bitcoin e qual è il suo significato o il suo utilizzo, comparato alla moneta classica. E poi vedremo come fare a rendere il nostro PC-on-a-Board (alias Raspberry Pi) un elemento attivo della blockchain, facendolo diventare un instancabile minatore digitale. Proveremo, inoltre, a calcolare quanto costa "minare" un Bitcoin.

BLOCKCHAIN E BITCOIN

Cerchiamo di capire come funziona la blockchain attraverso un confronto con le grandezze e i concetti economici che già esistono e con i quali abbiamo una certa familiarità. Successivamente potremo entrare nel dettaglio della tecnica che genera i Bitcoin. Il Bitcoin è una moneta virtuale il cui nome deriva dall'unione di *coin* (moneta) e *bit*; può essere utilizzato a livello globale e, come ogni altra moneta, può essere soggetto a conversioni di valuta.

Creato nel 2008 da una persona conosciuta con lo pseudonimo di Satoshi Nakamoto e lanciato ufficialmente nel 2009, il Bitcoin è arrivato a superare il valore dell'oro ed ha addirittura oltrepassato i 17.000 dollari nel dicembre del 2017.

Uno degli aspetti interessanti del Bitcoin è che con la sua nascita è stato introdotto un sistema monetario che opera in modo differente da quelli tradizionali e che si basa sulla decentralizzazione e sull'anonimato.

Delle due caratteristiche, la prima implica che il sistema monetario sottinteso dall'adozione del Bitcoin, non preveda l'istituzione e quindi l'utilizzo di alcun organismo di controllo centrale, mentre le monete convenzionali sono gestite da una Banca Centrale, che è un organismo di controllo che le emette e garantisce che non siano false o contraffatte, ma anche che il cambio sia coerente. Il Bitcoin non ha bisogno di un organismo di controllo centralizzato e il motivo di ciò sarà chiarito tra breve. La caratteristica dell'anonimato, invece, implica che l'identità di coloro che utilizzano il Bitcoin non venga mai rivelata, al contrario di quanto avviene con le valute tradizionali.

Pertanto l'anonimato è garantito dal fatto che il Bit-

coin è una valuta crittografata. I pagamenti in Bitcoin avvengono con soluzioni basate su blockchain, ovvero una tecnologia che consente il passaggio di informazioni all'interno di una rete di computer tra i nodi che la compongono. Le caratteristiche che rendono la blockchain così interessante sono la sicurezza, la trasparenza e l'anonimato.

La questione dell'anonimato è probabilmente la più controversa, soprattutto per via del fatto che l'economia globale sta cercando di arginare fenomeni di criminalità diffusa, riciclaggio del denaro, traffico illecito, vendita di armi, droghe e sostanze pericolose, nonché altre problematiche alimentate dall'utilizzo del denaro contante, il cui scambio non è tracciabile. Parimenti, il Bitcoin non consente il tracciamento, ragion per cui è timore diffuso quello che possa essere un veicolo privilegiato di riciclaggio; di tali questioni imprescindibili, che esulano da questa trattazione, si stanno occupando studiosi, giuristi, politici, sociologi, tecnici, giornalisti ed economisti. Come molte altre tecnologie di facile utilizzo ed accessibilità, anche quella delle criptovalute si è diffusa molto velocemente e può essere utilizzata da chiunque scaricando e installando semplici applicativi software o collegandosi a determinati siti web, che consentono di inviare, ricevere e memorizzare Bitcoin alla stregua di un vero e proprio portafoglio online. Ciò che però preoccupa è che gli operatori di questi servizi non figurano tra i soggetti "coperti" dalle normative antiriciclaggio.

Dal punto di vista economico, di tempo ne è passato dal primo sistema finanziario in cui "1 g d'oro valeva il suo controvalore in oro"; la golden share è stata eliminata e il controvalore sembra un concetto superato, anche grazie al numero e al tipo di diversi prodotti finanziari attualmente disponibili sul mercato. Il sistema economico mondiale si sta, infatti, smaterializzando sempre di più: questo dipende sia dal fatto che le economie sono diventate di servizio, sia dal superamento del concetto di produzione industriale e materiale come unico mezzo per poter stabilire che esista qualcosa cui dare un prezzo. Quando andiamo in giro per negozi, accettiamo l'idea di acquisire un bene materiale in cambio di carta con delle figure convenzionali disegnate sopra, ossia le banconote: ciò perché siamo tutti d'accordo che quel pezzo di carta abbia un determinato valore. Accettiamo anche il concetto che un negoziante debba recepire questi pezzi di carta come forma di moneta di scambio.

Accettiamo altresì che le monete non siano tutte uguali, che esistano dei tassi di cambio, che tutte abbiano un valore intrinseco, che possano sostituire

l'oro per via del fatto che sono più facilmente utilizzabili, che le monete in circolazione siano tutte autentiche perché c'è un organo di controllo che lo verifica.

Tutti questi assunti sono figli di una serie di presupposti che tengono in piedi il sistema economico così come lo conosciamo oggi.

Di fatto la moneta convenzionale è un'obbligazione, emessa dalla Banca Centrale dello Stato e che si impegna a fornire il controvalore a chi la restituisce alla stessa banca.

Il Bitcoin è fuori da questa logica, quindi in molti si fanno domande come: "quanto vale un Bitcoin?", "Se compro un Bitcoin che cosa ho in mano?", "Dove posso spendere un Bitcoin?", nonché la sua versione semplificata "Posso comprare il pane con Bitcoin?".

Sono domande che strappano un sorriso per la loro semplicità e non c'è da stupirsi, perché l'attuale sistema economico è stato concepito affinché fosse fruibile anche dalla persona più incolta. Ma siccome nel futuro la moneta potrebbe essere virtuale e digitale, è opportuno spiegare il meccanismo, non certo semplice come quello del denaro convenzionale, che le sta alle spalle e che passa dalla blockchain. Tutti quelli che hanno un conto corrente, indipendentemente dal numero di movimenti, sanno che in ogni momento è possibile controllare il saldo, quali sono state le transazioni effettuate, sia in ingresso sia in uscita, e tutti gli estremi dettagliati dei destinatari del denaro che non è più disponibile. La banca è legalmente responsabile di memorizzare tutte queste informazioni all'interno di un sistema privato e protetto. Deve altresì consentire imperativamente l'accesso ubiquo a tali informazioni ai propri clienti, in qualsiasi momento.

L'accesso a queste informazioni è consentito soltanto ai titolari del conto e agli operatori abilitati della banca stessa.

Questi criteri garantiscono, con assoluta certezza, e per tutti i soggetti coinvolti, che i movimenti siano riservati, privati, tutelati ed autentici in qualunque istante di tempo.

Fin qui tutto bene; ora proviamo ad immaginare un database che possa contenere un numero infinito di elementi.

Supponiamo, inoltre, che ogni volta che la banca scrive un nuovo record all'interno del database, definisca la prima transazione, di seguito la seconda e così via dicendo, rispettivamente utilizzando il codice "1", "2", "3", ecc.

Dopo un certo numero di transazioni, la banca, sui suoi database, inizierà a scrivere il record "i" ed i

successivi "i+1", "i+2" e così via dicendo.

Quanto detto consente di fare due considerazioni:

- non è possibile scrivere una transazione con indice precedente perché la scrittura dei record all'interno del database è univoca e avviene in un solo verso;
- se in un certo lasso di tempo la banca sta scrivendo il numero "k", la corrispondente transazione non può ancora essere ritenuta valida; ciò è analogo, in qualche modo, alla differenza che c'è fra saldo contabile e saldo disponibile, quindi qualcosa che per noi è familiare; le operazioni vengono contabilizzate nel tempo necessario a che ciò avvenga e ciò implica che tutte le transazioni precedenti "k" possono essere consultabili, mentre quest'ultima no.

Un'altra considerazione riguarda il costo della transazione, vale a dire la commissione, giustificata dalla copertura dei costi di esercizio dell'attività bancaria; l'importo di questa commissione non è fisso ed è, anzi, soggetto alle regole del libero mercato, per cui ogni banca può decidere il proprio. Rimanendo sul tecnico, il Bitcoin può essere facilmente concepito come una serie di stringhe di codici univocamente definiti che hanno valore soltanto perché accettati come moneta di scambio tra gli utenti; i codici in oggetto funzionano come dei veri e propri conti correnti.

Più che un meccanismo di scambio basato su una moneta, si può pensare a un sistema ibrido: una vera e propria ridefinizione del baratto, ma in chiave digitale.

CENTRALIZZATO VS DECENTRALIZZATO

Per comprendere al meglio il concetto della decentralizzazione, la strada più semplice, veloce ed efficace è quella di analizzare il concetto di sistema monetario "centralizzato": quando un sistema economico è governato da un'autorità centrale, il sistema si dice "centralizzato". Gli aspetti più importanti sono la stampa delle banconote, il conio della moneta, la riscossione delle tasse, la regolamentazione delle banche e del credito più in generale.

Le leggi e le regole che si occupano di indirizzare le attività in tal senso sono emanate dal governo.

Tutto ciò che riguarda ogni singola transazione deve essere approvato e memorizzato al fine di evitare ogni genere di possibile controversia futura. In tal senso, le banche operano per garantire l'unicità e l'univocità delle transazioni, in termini sia di scambio che di attori dello scambio stesso.

In un sistema decentralizzato, invece, tale struttura

non esiste, tuttavia le necessità del sistema finanziario restano invariate; si pone quindi il problema di chi dovrebbe garantire la correttezza formale e sostanziale dello svolgersi delle operazioni di un sistema senza un ente regolatore.

Prima di tutto, è importante specificare che le normative in ambito economico non sono trasversali: differiscono da nazione a nazione. Il sistema Bitcoin risponde alla comune esigenza di trovare un metodo per la generazione della moneta.

È chiaro che un sistema centralizzato prevede alcuni meccanismi che non possono essere alterati in alcun modo.

Ma cosa cambia in un sistema decentralizzato, concretamente? Ebbene, la decentralizzazione non altera la necessità di memorizzare ogni transazione, ma viene da chiedersi come si fa a farlo senza un organo centrale adibito a tale scopo, chi garantisce che la memorizzazione sia attendibile e che il dato sia rispondente al vero? In breve, anche se c'è un dato memorizzato, come si fa ad essere sicuri che sia vero?

Per spiegarlo immaginiamo che ci siano due utenti, Alice e Bartolo e che Alice voglia inviare del denaro a Bartolo: nel sistema bancario c'è un gruppo di persone che verifica la disponibilità del credito di Alice ed è incaricata di raccogliere le informazioni relative alla transazione verso Bartolo, come ad esempio il numero di conto di Bartolo e la cifra da trasferire.

Degli utenti adibiti al controllo delle transazioni, almeno uno riceverà la notizia che Alice vuole trasferire del denaro: supponiamo si tratti di Chiara, la quale per poter guadagnare la commissione sull'operazione, aprirà la transazione, ovvero creerà un nuovo record all'interno del database. Ricordiamoci che il database è condiviso tra tutti gli impiegati, pertanto non appena la transazione sarà avvenuta, anche gli altri utenti (che chiamiamo Daniela, Ernesto, Francesca ecc.) sapranno che lo è.

Chiara deve aggiungere la transazione al database per essere sicura che l'operazione possa essere numerata; così facendo, garantirà che il database contenga sempre informazioni aggiornate, uniche e consistenti. Quando tutti gli impiegati verificano che la transazione è stata inserita all'interno del database, essa viene dichiarata valida e quindi a Chiara viene riconosciuta la commissione.

Se Daniela o Francesca dovessero cercare di inserire nel database la stessa transazione, sarebbe comunque impossibile per loro completare l'operazione. Se Chiara e Daniela dovessero cercare di processare contemporaneamente la stessa transazione tra Alice

e Bartolo, solo una delle due potrebbe completare l'operazione per prima: quella più veloce.

Questo caso potrebbe rappresentare una qualunque persona che utilizzi un'applicazione di portafoglio elettronico su un dispositivo, sia esso un PC oppure uno smartphone o un dispositivo portatile in generale. L'applicazione non è altro che un software capace di memorizzare i Bitcoin, mantenendo in memoria una traccia degli indirizzi IP dei "minatori" (coloro che operano nella rete del Bitcoin); a questi ultimi, tramite i loro indirizzi IP, viene notificata ogni richiesta di trasferimento di denaro. In pratica, ogni volta che creiamo una nuova transazione, essa viene opportunamente notificata.

Inoltre l'applicazione interroga periodicamente i "minatori" per verificare se è stato ordinato il trasferimento di denaro a nostro beneficio, ovvero se abbiamo ricevuto qualche pagamento, il che ci introduce un altro concetto: tutte queste operazioni sono calcoli e vengono effettuate tramite l'utilizzo di un computer. Siamo abituati a concepire che più è veloce il computer, più operazioni "contemporaneamente" può eseguire.

Questo è un nuovo elemento molto importante: il sistema di gestione delle transazioni su cui è basato il Bitcoin è concorrenziale e si basa sul fatto che la capacità computazionale fa la differenza.

Il sistema è comunque molto attento alle differenze di partenza: supponiamo che Chiara e Daniela partano nello stesso momento a cercare di risolvere la transazione ed inserirla nel database; il modo in cui è possibile ottenere che tutti concordino su chi abbia processato la transazione e quindi chi meriti le commissioni, è la risoluzione di un puzzle, sotto forma di quesito matematico.

Per ogni transazione che può essere processata da ciascuno degli impiegati di questa "banca", viene generato un quesito matematico, molto difficile da risolvere, e molto più complesso, quindi, di un'operazione elementare come la somma o il prodotto. L'algoritmo che genera il problema matematico lavora in modo casuale per creare problemi sempre diversi e di difficoltà sempre crescente rispetto a quelli generati fino all'istante precedente; quindi il sistema diventa sempre più complesso e il problema matematico diviene sempre più difficile per cui inserire transazioni all'interno del record richiede impiegati sempre più veloci e macchine sempre più potenti. In ogni caso, sappiamo per certo una cosa: il problema verrà risolto; è solo questione di tempo! Ma Bartolo, ora che ha avuto questa quantità di moneta, cos'ha ottenuto effettivamente? Non è carta colorata con figure e numeri, quindi di che cosa

si tratta? La particolarità dell'era del Bitcoin è che non abbiamo alcuna difficoltà a immaginare che un conto corrente sia la controparte virtuale di un certo mucchio di banconote su un tavolo, ma ci spaventa il discorso che la moneta sia interamente virtuale. Eppure esiste un termine di paragone simile a quanto avveniva in precedenza: il cosiddetto mining. Come in tempi antichi si andava nelle miniere per cercare di estrarre materiali preziosi, oggi come oggi l'attività di creazione dei Bitcoin prende il nome di mining.

I moderni minatori lavorano utilizzando il meglio della loro potenza di calcolo proprio per risolvere quei problemi matematici complessi di cui parlavamo in precedenza.

Il Bitcoin è, quindi, il risultato della risoluzione del problema matematico di cui sopra.

Lo sforzo dei minatori digitali è quello di investire tempo, energia elettrica e denaro nella ricerca innanzitutto di macchine in grande quantità, che possano essere aggiornate, in termini di capacità computazionale, e che lavorino costantemente.

I minatori lavorano, quindi, sia attraverso le commissioni relative alle transazioni che vengono processate ed inserite all'interno del database sia con ogni record inserito.

Ciascuno di essi, infatti, concorre alla risoluzione del problema matematico di cui sopra e, pertanto, è fondamentale che la conoscenza di tutto ciò che c'è all'interno del database sia disponibile per tutti e aggiornata in tempo reale.

Una volta che i minatori sono stati notificati circa una transazione, essi la processeranno cercando una soluzione al puzzle ad essa associato. Il minatore che troverà per primo la soluzione al puzzle si guadagnerà la commissione, oltre a un ammontare fisso di denaro (in Bitcoin, o frazioni di esso).

È TUTTO CHIARO?

Fino a questo momento avrete certamente diversi dubbi. Vi starete probabilmente chiedendo se è possibile passare la vita a minare Bitcoin senza riuscire a risolvere mai una volta uno solo degli indovinelli o problemi matematici proposti. Ebbene, la risposta è che può succedere; esattamente come, nella vita di tutti i giorni, potete essere gli ultimi in coda al botteghino, secondi alla maratona, oppure come può accadere che non vinciate mai la lotteria. Sicuramente avete anche tutta una serie di dubbi tecnici e ancora non avete capito come si può arrivare ad avere un Bitcoin. Inoltre, in questa fase il mining è ancora soltanto un racconto, ma niente di concreto.

Però non scoraggiatevi, perché questa lunga descrizione introduttiva è stata fondamentale perché prima di poter metterci a fare cose, scrivere codice, scaricare pacchetti, configurare macchine o qualunque altra cosa di pratico, è fondamentale capire qual è lo scenario operativo.

Ora che tutte queste cose sono finalmente chiare, possiamo cominciare: vediamo di dare a tutti gli "oggetti" sopra menzionati i nomi corretti scelti dalla comunità Bitcoin: l'insieme di transazioni (processabili da ogni minatore) viene chiamato "block", e un insieme di transazioni viene perciò detto "blocco" di transazioni. Il database comune e condiviso viene chiamato "block-chain", dal momento che si tratta di una catena di blocchi ordinati. Ciascuno di essi contiene un insieme di transazioni già processate e validate. Il puzzle di elevata complessità che i minatori devono cercare di risolvere è, in sostanza, rappresentato dall'output prodotto da una funzione di hash crittografico.

Inoltre, dal momento che ai minatori è richiesto di eseguire un numero elevato di tentativi al fine di trovare una soluzione al puzzle, si dice che essi stanno dimostrando ("provando") di avere lavorato duramente; in inglese questo termine prende il nome di Proof of Work, o semplicemente PoW.

BITCOIN IN PRATICA

Adesso che vi abbiamo trasmesso il background del Bitcoin, vi spieghiamo come entrare a far parte della blockchain, direttamente da casa vostra. In particolare, supponiamo di utilizzare, come computer, una scheda della famiglia Raspberry Pi e di utilizzare il classico sistema operativo Raspbian.

Per iniziare il mining bisogna innanzitutto aggiornare il sistema operativo: essendo una distribuzione Linux Debian-based, Raspbian si basa, per la gestione dei pacchetti, su Aptitude (ammesso che non lo abbiate cambiato). L'aggiornamento si può effettuare con i seguenti comandi:

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

Quando Aptitude vi chiederà conferma di quanto aggiornare, confermate e attendete la fine del pro-

```
[2015-06-25 02:36:40] Started cgminer 4.9.0
[2015-06-25 02:36:40] -----start voltage-----
[2015-06-25 02:36:41] Send voltage aab00c0
[2015-06-25 02:36:41] Send voltage ok
[2015-06-25 02:36:41] -----start freq-----
[2015-06-25 02:36:41] Send frequency 82128614
[2015-06-25 02:36:42] Send freq getstatus 84000400
```

Fig. 2 - Output su terminale all'avvio di cgminer.

Tabella 1 - Specifiche delle schede Raspberry Pi.

	Raspberry Pi				
	V1 Model A	V1 Model B+	V2 Model B	V3 Model B	V3 Model B+
SoC	Broadcom BCM2835		Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2837B0
CPU	700 MHz single-core ARM1176JZF-S		900 MHz 32-bit quad-core ARM Cortex-A7	1.2 GHz 64-bit quad-core ARM Cortex-A53	1.4 GHz 64-bit quad-code ARM Cortex-A53 (ARMv8)
GPU	Broadcom VideoCore IV OpenGL ES 2.0 1080p30 H.264 high-profile decoder ed encoder 250 MHz per BCM2835 e BCM2836. 400 MHz dual core per BCM2837 (1080p60)				
Memoria (SDRAM)	256 MB (condivisa)	512 MB (condivisa)	1 GB (condivisa) LPDDR2 (450 MHz)	1 GB (condivisa) LPDDR2 (900 MHz)	1 GB (condivisa) LPDDR2
Memoria	SD / MMC / SDIO card slot	microSD			
Collegamenti di rete	Nessuno	Ethernet 10/100 Mbit/s		Ethernet 10/100 Mbit/s, WLAN 802.11n, Bluetooth 4.1	Ethernet 10/100 Mbit/s, WLAN 802.11b/g/n, Bluetooth 4.2
Corrente (potenza) assorbita	300 mA (1.5 W)	600 mA (3,0 W)	800 mA (4,0 W)		
Tensione di alimentazione	5V				

cesso. Poi dovete installare i pacchetti aggiuntivi e le relative dipendenze, con il comando:

```
sudo apt-get install libusb-1.0-0-dev libusb-1.0-0
libcurl4-openssl-dev libncurses5-dev libudev-dev screen
libtool automake pkg-config libjansson-dev screen
```

A questo punto scaricate cgminer:

```
git clone https://github.com/bitmaintech/cgminer
```

Ora non resta che spostarci nella cartella di cgminer ed eseguire alcune, semplici, operazioni per effettuare la configurazione necessaria:

```
cd cgminer
sudo ./autogen.sh
export LIBCURL_FLAGS='-I/usr/include/curl'
sudo ./configure --enable-bmnc
sudo make
```

Se tutto è stato fatto correttamente, siamo pronti a partire e a questo punto basta inserire la seguente stringa di testo:

```
sudo ./cgminer --bmnc-options 115200:0.57 -o XXXXX -u
YYYYY -p ZZZZZ --bmnc-voltage 0800 --bmnc-freq 1286
```

in cui:

- XXXXX è il POOL address;
- YYYYY è il vostro USERNAME;
- ZZZZZ è la vostra password.

Fatto questo siete operativi, però tenete presente che in caso di spegnimento della Raspberry Pi, causato da un blackout o da un calo della tensione di alimentazione sotto il valore minimo, sarebbe necessario provvedere al riavvio manuale della routine, perché alla riaccensione riparte solo Raspbian. Per ottenere il riavvio automatico della routine è indispensabile modificare il funzionamento della scheda, programmandola per eseguire in automatico l'avvio di cgminer. Allo scopo, modificate il file **rc.local** in maniera opportuna, ricordando che è localizzato nella cartella **/etc** e che le modifiche andranno fatte da superutente (**sudo**).

QUANTO COSTA MINARE UN BITCOIN?

Le operazioni che avete letto nella sezione precedente sono state eseguite su cinque schede diverse della famiglia Raspberry, ossia:

- Raspberry Pi 1 model A;
- Raspberry Pi 1 B+;
- Raspberry Pi 2 B;
- Raspberry Pi 3 B;
- Raspberry Pi 3 B+.

Esse sono differenti tra loro in termini di RAM disponibile, ma anche dal punto di vista della potenza computazionale, molto limitata sulla versione 1 ma decisamente maggiore a partire dalla 2 B+ in poi. Le prestazioni e i consumi sono elementi determinanti nel valutare quanto costa guadagnare un Bitcoin, giacché il mining richiede la costante attività

di un computer in rete. Nella **Tabella 1** vengono riassunte le caratteristiche tecniche delle schede testate, rilevanti ai fini della valutazione; nelle prove, le schede sono state alimentate da un alimentatore AC/DC da 5 V, 2,4 A.

Dal punto di vista del consumo energetico, i valori rilevati sono riportati nella **Fig. 3**; quelli a riposo sono relativi al solo avvio del sistema operativo. Ora cerchiamo di capire meglio quanto questo costa minare Bitcoin; partendo dai dati sui consumi elettrici, convertiamo i mAh in Ah, semplicemente dividendo i valori per 1.000.

Per esempio, 310 mAh corrispondono a 0,31 Ah; per ottenere il consumo corrispondente in Watt/ora [Wh], serve moltiplicare per il valore di tensione in uso. Nel nostro caso, sono 5 V, ovvero la tensione operativa della nostra Raspberry Pi. Quindi, in Wh, stiamo consumando $0,31 \times 5 = 1,55$ Wh.

Per capire quanto costa "produrre" i Bitcoin mancano ancora due dati, perché dobbiamo sapere:

1. quanto costa l'energia che consumiamo;
2. quanto tempo ci vuole per minare un Bitcoin.

La prima informazione è abbastanza facile da reperire: basta controllare il contratto di fornitura della rete elettrica. Se siete collegati al provider nazionale dell'energia elettrica in Italia, facilmente avrete tariffe simili a quelle in **Tabella 2**.

Per semplicità, immaginiamo di arrotondare il costo a 0,05 €/kWh, ovvero 0,00005 €/Wh. Ipotizziamo, inoltre, che la fascia oraria non abbia influenza, ovvero che il prezzo sia uniforme. In queste ipotesi, avremmo $1,55 \times 0,00005 = 0,000075$ €. Bene, questo è il costo di un'ora di lavoro per il nostro sistema Raspberry Pi che mina Bitcoin.

Resta da rispondere alla seconda domanda, ma qui

le cose si complicano e non poco, perché purtroppo, dati i meccanismi spiegati in precedenza, la crescita sempre più importante del peso computazionale del mining, l'incremento della complessità del problema e l'eterogeneità delle macchine coinvolte, non è possibile prevedere quante ore dovrete tenere il vostro sistema on-line per riuscire ad aggiungere un blocco alla blockchain.

Vale la pena di notare anche che, da questa stima preliminare, sono rimaste fuori alcune voci di costo, considerate minori, ma che in effetti sono tutt'altro che trascurabili.

Prima tra tutte, infatti, anche se sembrerà banale, il costo della scheda, che comunque, nella sua ultima versione disponibile è ancora al di sotto dei 50 €. Poi c'è il costo della connessione a Internet e dell'elettricità consumata dal router. C'è inoltre la resa dell'alimentatore della Raspberry Pi, nonché il decremento delle prestazioni della scheda dovuto all'aumento della temperatura operativa del SoC durante le operazioni di mining; a quest'ultimo

Tabella 2 - Costi tipici dell'elettricità in Italia.

Fascia Oraria	Prezzo Energia
F0 (Fascia unica)	0,05090 €/kWh
F1 (Giorno)	0,05475 €/kWh
F2 e F3 (sera e weekend)	0,04908 €/kWh

fattore è possibile ovviare tramite l'utilizzo di dissipatori passivi sul processore, ma se utilizzaste dei sistemi di raffreddamento attivi, i consumi si crescerebbero ancora.

Dal punto di vista computazionale, vediamo qual è la percentuale di utilizzo della CPU per ciascuna delle board; contestualmente riportiamo, grazie ad un grafico dedicato, l'occupazione di memoria RAM. I valori riportati in **Fig. 4** e in **Fig. 5** dimostrano come sia davvero difficile utilizzare le schede per fare altro mentre sono a lavoro per minare Bitcoin. Vale la pena di notare che i valori riportati individuano le voci di costo e di performance di una singola scheda. Navigando attraverso forum, siti, blog di informazione e altro, trovate, anche abbastanza spesso, immagini di serie di schede, in particolare proprio Raspberry Pi, di diversa generazione, collegate ad uno switch di rete e al lavoro per minare Bitcoin. È intuitivo che le considerazioni lette fino ad ora valgono per ciascuna di quelle schede. Il fatto che siano simultaneamente al lavoro non diminuisce il consumo energetico di ciascuna. Ed allora, probabilmente la vostra domanda di-

Consumo di corrente [mAh]

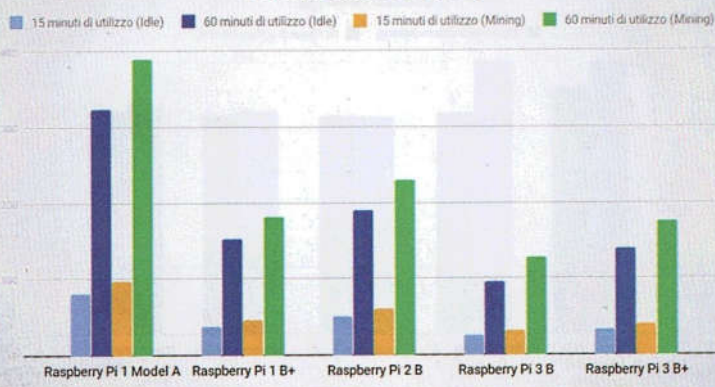


Fig. 3 - Costo dell'energia utilizzata per minare Bitcoin.



ATTENZIONE!

In questo articolo abbiamo preso l'esempio del Bitcoin per spiegare, attraverso un esempio tangibile perché di grande attualità, come funziona la blockchain. La trattazione ha fini meramente tecnici e divulgativi e non vuole in alcun modo essere un incentivo all'investimento finanziario in Bitcoin, che abbiamo esposto con neutralità. Anche l'esempio applicativo basato sulla realizzazione di un server per mining ha la sola finalità di spiegare come sia possibile entrare nel mondo del Bitcoin con risorse hardware limitate, esponendone pro e contro.

venta: perché metterne tante al lavoro tutte insieme, nella stessa rete? Ebbene, perché mettere più schede al lavoro per fare il mining è come comprare più biglietti della lotteria: le probabilità di "vincita" sono maggiori. Se tutto il mondo minasse Bitcoin, contemporaneamente, la crescita della complessità del problema di cui abbiamo parlato sarebbe molto più rapida. Quali sono, quindi, i margini di miglioramento del profitto e delle prestazioni del sistema che abbiamo appena visto? Il primo punto è sicuramente l'hardware. L'ideale sarebbe avere a disposizione un'architettura quantistica, ma questo tipo di soluzione non è assolutamente disponibile a livello commerciale ma, anche soltanto a livello teorico, sicuramente risolve il problema di avere calcolatori di potenza superiore. Il secondo punto riguarda l'implementazione, che potrebbe essere ottimizzata e resa scalabile a seconda della macchina sulla quale vengono eseguiti i calcoli. Qualcosa di simile a quello che avviene con i motori grafici dei videogiochi, grazie alle cui impostazioni possiamo scegliere il livello di dettaglio e la risoluzione della rappresentazione dell'immagine per poter eseguire lo stesso gioco anche su macchine più datate.

CONCLUSIONI

Questo articolo è nato con la chiara intenzione di

trasformare dei concetti economici, oggettivamente, a volte, molto complicati, in qualcosa di più semplice. Chiariti i presupposti teorici, è stato incredibilmente più immediato comprendere cosa la tecnica abbia creato. Siamo di fronte all'inizio di una nuova era economica che ha, come fondamento, la digitalizzazione spinta del concetto stesso di scambio. Dopo aver analizzato questo scenario futuribile, abbiamo visto e toccato con mano nel dettaglio che cosa può voler dire utilizzare una moneta virtuale, provare a diventare parte della blockchain, ma anche minare un Bitcoin. Tutto ciò a partire da una piattaforma hardware a basso costo, ovvero Raspberry Pi. Abbiamo visto i passaggi fondamentali per rendere operativa la scheda, analizzato le sue prestazioni e proiettato, sul lungo periodo, il costo di un Bitcoin, allo scopo di farci un'idea di quanto possa essere sostenibile economicamente, per un privato, minare un Bitcoin e, conseguentemente, del perché esistono interi stabilimenti all'interno dei quali si lavora con una mole impressionante di calcolatori ad altissime prestazioni solo al fine di creare blocchi aggiuntivi per la blockchain. D'altronde la blockchain non è più qualcosa di lontano dal settore finanziario, perché già in maggio scorso è avvenuta, tra due banche (HSBC ed ING), la prima transazione basata su di essa e ciò potrebbe avviare l'impiego su larga scala. ■

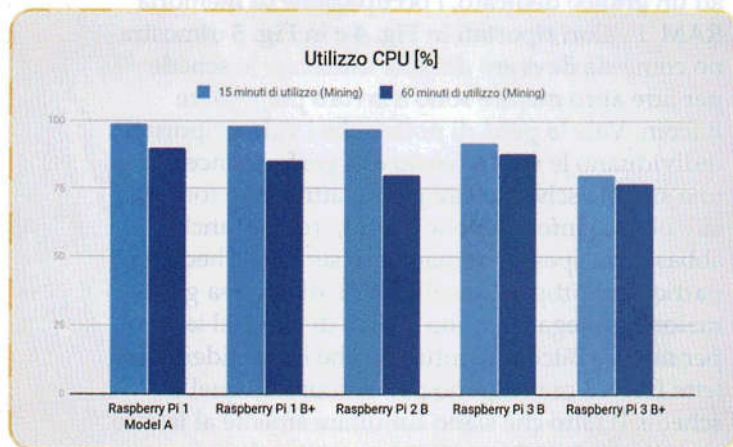


Fig. 4 - Utilizzo della CPU nel tempo.

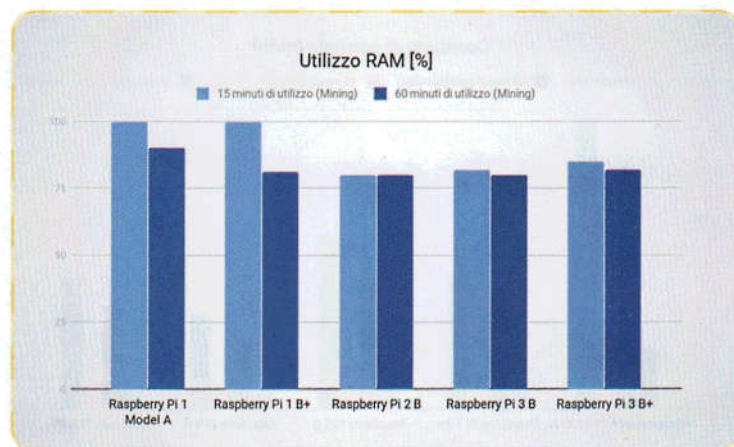


Fig. 5 - Utilizzo della RAM nel tempo.

MICRO:BIT SMART CAR

Piattaforma multifunzione costruita attorno alla scheda didattica micro:bit della BBC.

Integra un simpatico robot, un kit per chi comincia ed anche una board per sperimentare.



di MARCO MAGAGNIN

Tempo fa, nel numero 213 (Marzo 2017) di Elettronica In abbiamo recensito il microcontrollore micro:bit, progettato dalla BBC come un oggetto finalizzato a sviluppare le capacità logiche, e le competenze, nel campo della programmazione dei computer, nei giovani che frequentano i primi cicli della scuola.

L'iniziativa è stata coronata da successo e diversi produttori hanno sviluppato e commercializzato dispositivi aggiuntivi per ampliare le possibilità offerte dal micro:bit.

La Smart Car della YahBoom ci ha particolarmente incuriosito, in quanto permette di sperimentare un'ampia gamma di situazioni tipiche di ciò che viene definito "physical computing" utilizzando un

unico kit di costo contenuto.

Tra le opportunità di sperimentazione, per quanto riguarda il micro:bit utilizzato singolarmente, possiamo elencare:

- la gestione di una matrice di LED 5 x 5;
- l'utilizzo di un accelerometro in grado di rilevare e riconoscere diversi tipi di movimento come lo scuotimento, le rotazioni e la caduta libera;
- l'utilizzo di una bussola elettronica in grado di rilevare l'orientamento e la rotazione in gradi rispetto al Nord. Un'ulteriore possibilità potrebbe essere l'utilizzo come rilevatore di metalli (magnetici);
- l'utilizzo di due pulsanti sia singolarmente che premuti in contemporanea;

- l'utilizzo del modulo Bluetooth Smart Technology che permette di connettere il micro:bit con altri dispositivi dotati della stessa tecnologia, come altri micro:bit, PC, smartphone, kit Bluetooth, tablet e fotocamere;
- l'utilizzo del modulo a radio frequenza per la comunicazione tra due o più micro:bit;
- la misura della temperatura ambiente utilizzando il termometro integrato;
- la possibilità di comunicazione via bus I2C ed SPI.

L'associazione con la scheda di espansione permette le seguenti funzionalità aggiuntive:

- la comunicazione di comandi per mezzo di un telecomando ad infrarossi;
- la generazione di note e melodie musicali utilizzando il buzzer cablato sulla scheda di espansione;
- la gestione multicolore di due LED RGB da 10 mm;
- la gestione multicolore di ulteriori tre LED RGB;
- sulla scheda sono replicati i pin che fanno riferimento ai bus seriale, I2C ed SPI;
- sulla scheda sono inoltre replicati i pin delle periferiche del micro:bit non utilizzate dalla scheda di espansione. In particolare i pin P4, P5, P6, P7, P10;

- l'alimentazione da batteria agli Ioni di Litio con circuito di ricarica integrato nella scheda di espansione.

Le ulteriori funzionalità sono fruibili in modo ottimale assemblando completamente il kit "robot" che include i motori, le ruote ed il sensore ad ultrasuoni. In questa configurazione sono accessibili le ulteriori funzionalità:

- il controllo dei motori;
- l'utilizzo dei sensori frontali (agli infrarossi e agli ultrasuoni) per il controllo dell'ambiente circostante. Questi permettono di sperimentare funzionalità come evitare ostacoli e seguire oggetti;
- l'utilizzo dei sensori inferiori, agli infrarossi, per programmare un inseguitore di linea.

Infine, come ultima ciliegina, il tutto è controllabile per mezzo di una app per smartphone Android. In alternativa alla app è utilizzabile l'ambiente di sviluppo standard JavaScript Blocks Editor.

SCHEMA A BLOCCHI

In Fig. 1 riportiamo la descrizione dei pin del connettore del micro:bit mentre in Fig. 2 riportiamo il layout della scheda di espansione con l'indicazione dei dispositivi montati sulla scheda e la loro disposizione sulla stessa.

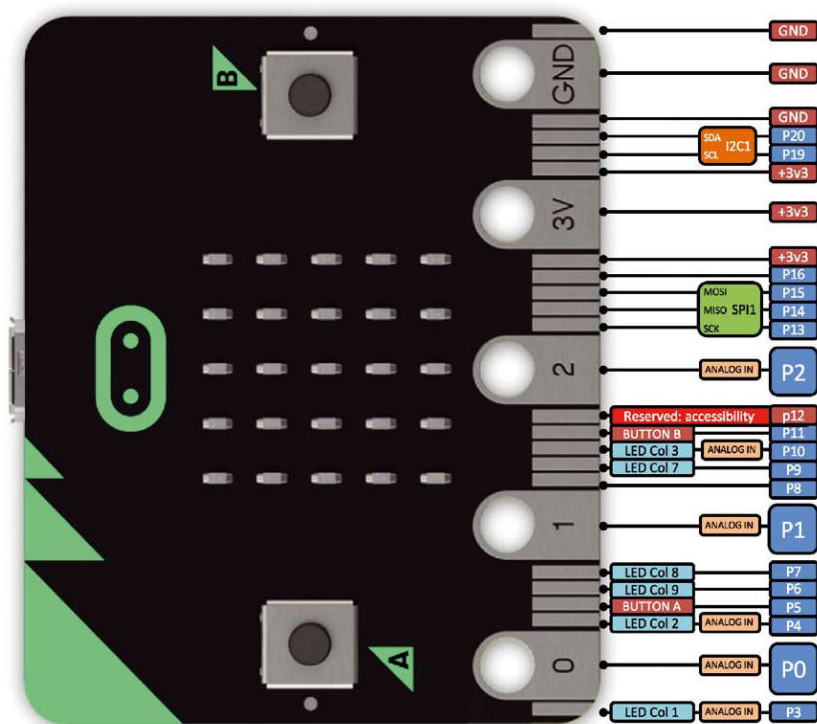


Fig. 1 - Pin disponibili sul connettore del micro:bit.

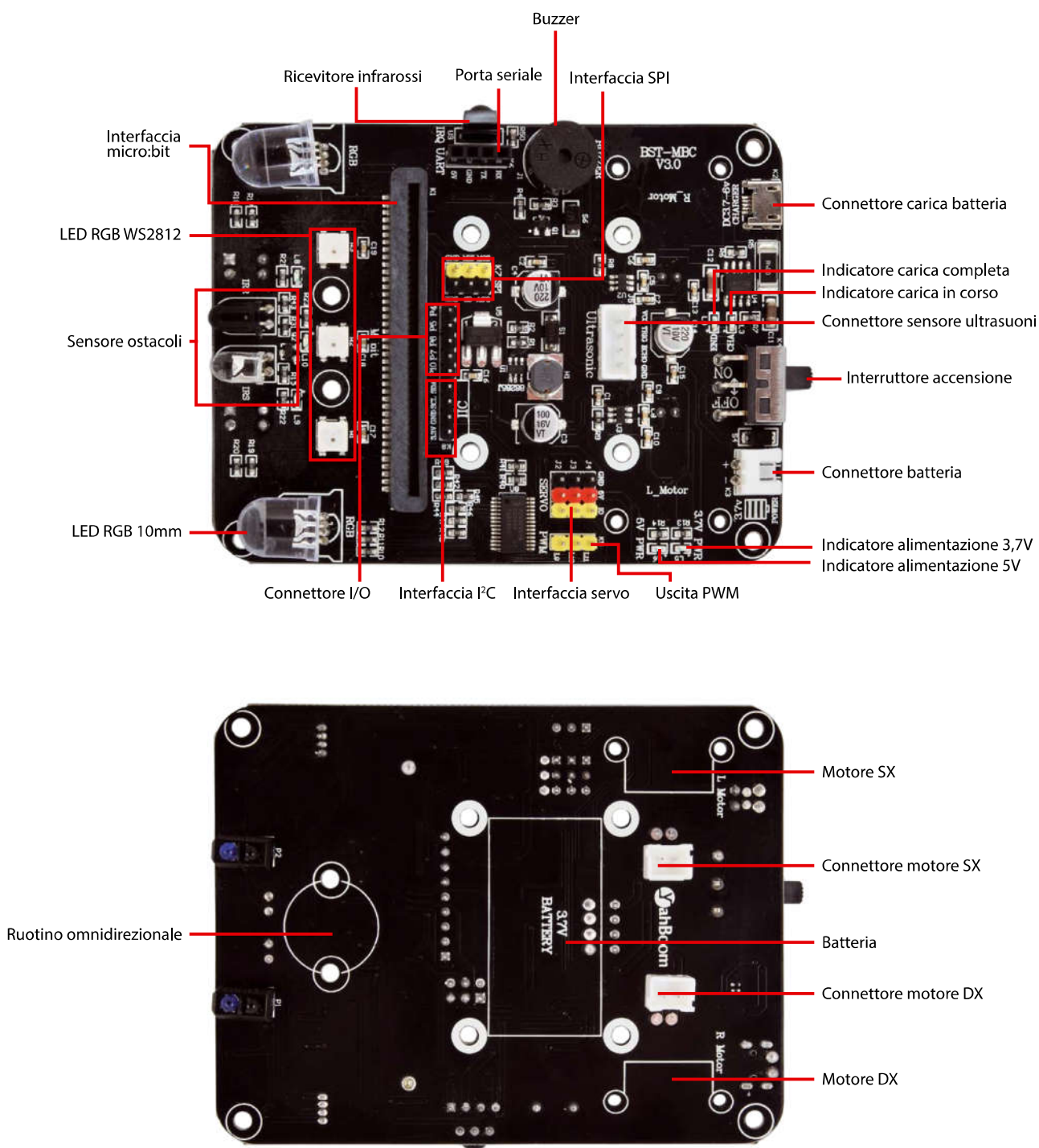


Fig. 2 - Viste superiore e inferiore della scheda di espansione Smart Car

Alcune note sulla scheda di espansione:

- il connettore per l'alimentazione e la ricarica della batteria non è utilizzabile per programmare il micro:bit. L'operazione di programmazione del micro:bit deve essere eseguita collegando il cavo

USB, proveniente dal PC, direttamente al connettore USB presente sul micro:bit;

- la ricarica della batteria è garantita dall'integrato LTC4056, appositamente progettato per la ricarica ed il mantenimento in carica degli elementi

agli Ioni di Litio. Lo stato di carica della batteria è indicato da due LED: il LED CHA (indicatore di carica in corso) rosso ed il LED END di colore verde che indica che la carica è terminata;

- L'integrato PCA9685, un driver PWM a 16 canali, pilotato tramite bus I2C, interfaccia tutti i device che richiedono un segnale PWM per essere pilotati, tra i quali i due motori in corrente continua, i due LED da 10 mm, i sensori di linea, le uscite per i servocomandi aggiuntivi e per il sensore all'infrarosso per evitare gli ostacoli. Nella **Tabella 1** diamo un'indicazione di quali pin dell'integrato PCA9685 fanno capo ai diversi sensori ed attuatori, mentre in **Tabella 2** sono elencati i collegamenti ai pin del micro:bit.

COSTRUZIONE DELLA SMART CAR

Tabella 1 - Collegamenti all'integrato PCA9685.

Pin PCA9685	Connessione
0	Led 10mm R
1	Led 10mm G
2	Led 10mm B
3	Servo 1
4	Servo 2
5	Servo 3
6	Sensore di linea SX
7	Sensore di linea DX
8	Sensore IR frontale
12	Mot. SX avanti
13	Mot. SX indietro
14	Mot. DX indietro
15	Mot. DX avanti

Tabella 2 - Collegamenti ai pin del micro:bit.

Pin micro:bit	Connessione
P1	Lettura analogica sensore Linea SX
P2	Lettura analogica sensore Linea DX
P3	Lettura analogica sensore IR frontale
P9	Trasmettitore IR frontale
P14	Impulso out sensore ultrasuoni
P15	Impulso in sensore ultrasuoni
P16	Neopixel Red
P17	Neopixel Green
P18	Neopixel Blu
P19	I2C SCL
P20	I2C SDA

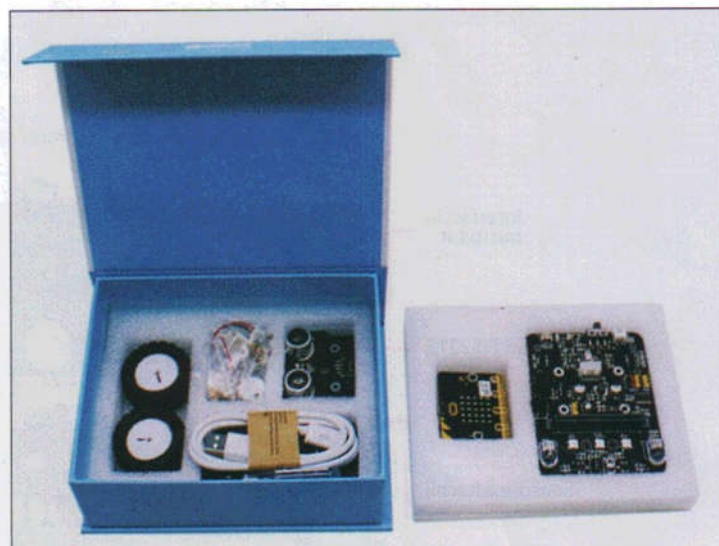


Fig. 3 - Contenuto della confezione della Smart Car.

Il piccolo robot didattico viene fornito in forma di kit parzialmente assemblato. La scheda di espansione è completamente montata mentre gli altri componenti devono essere montati prima dell'utilizzo. In questo modo non è necessario eseguire saldature. Tenete presente che nella confezione della Smart Car, distribuita da Futura Elettronica, è compreso il micro:bit.

La confezione, visibile in **Fig. 3**, contiene i seguenti componenti:

- 1 x scheda micro:bit;
- 1 x scheda espansione micro:bit;
- 1 x sensore a ultrasuoni con cavo di collegamento;
- 2 x motori;
- 1 x piastra di supporto per il sensore a ultrasuoni;
- 2 x pneumatici;
- 1 x cavo micro usb;
- 1 x controllo remoto a infrarossi;
- 1 x cacciavite;
- 4 x M3*40 colonnine in rame;
- 2 x M2*10 colonnine in rame;
- 2 x supporti sensore ultrasuoni;
- 1 x batteria al litio;
- 1 x manuale di istruzioni;
- 1 x ruotino omnidirezionale;
- 2 x supporti motore;
- viti e dadi.

Nella **Fig. 4** è visibile lo schema di montaggio che prevede le seguenti operazioni:

1. Montaggio dei motori utilizzando gli specifici supporti bianchi e le viti da 2x6mm e relativi dadi (viti dalla parte della scheda di supporto e

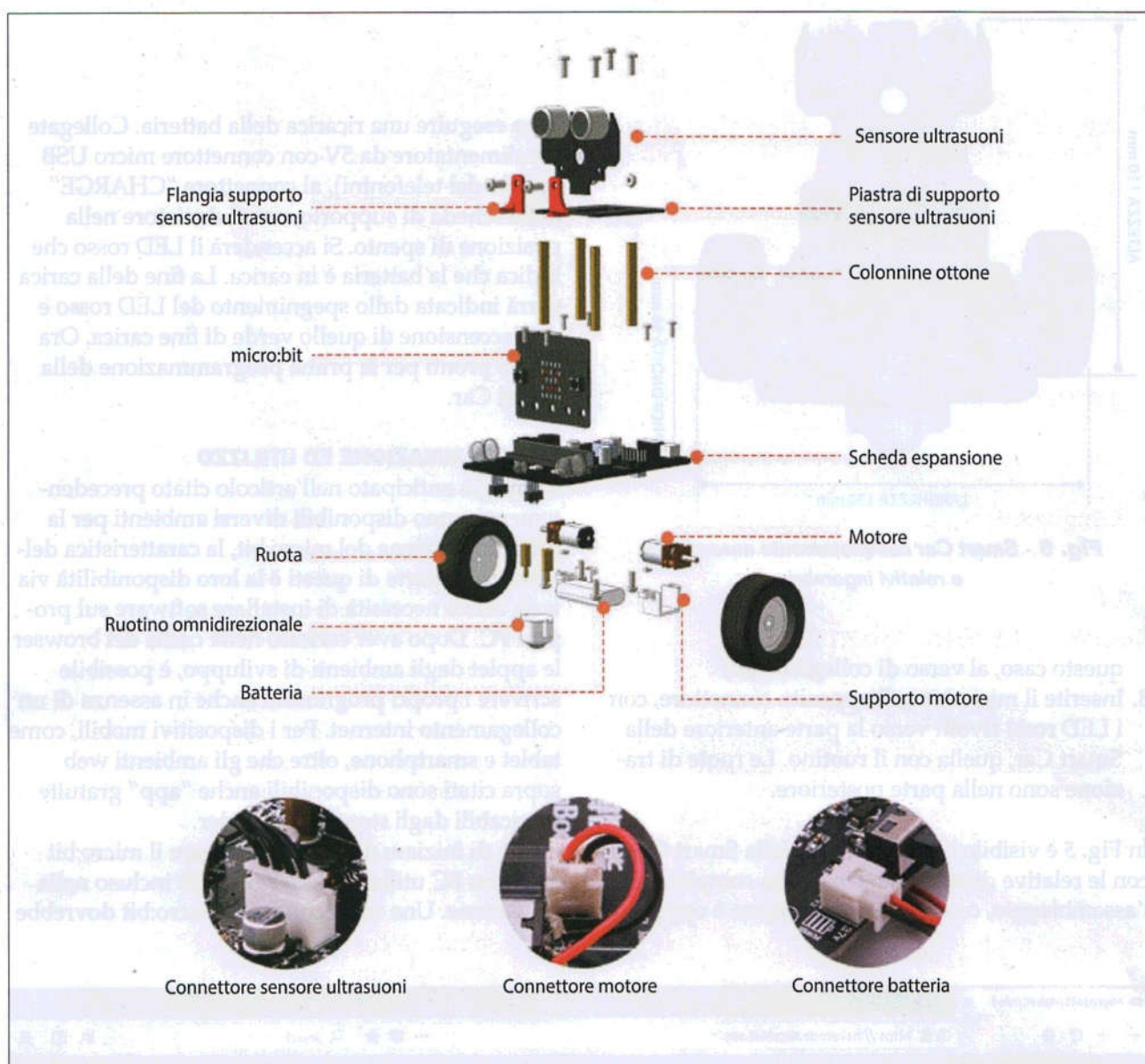


Fig. 4 - Schema di montaggio della Smart Car.

- dadi dalla parte dei supporti motore). Collegate i cavi dei motori agli appositi connettori a fianco degli stessi facendo attenzione al verso di collegamento;
2. Una volta montati i motori inserite le due ruote sugli assi dei motori esercitando la pressione necessaria (senza esagerare);
3. Posizionate il ruotino omnidirezionale e fissatelo con due viti da 2x6mm e relativi dadi (viti dalla parte della scheda di supporto e dadi dalla parte del ruotino);
4. Montate le quattro colonnine in ottone da 3x40mm che forniranno il supporto per il sensore ad ultrasuoni. Fissate le viti dalla parte del telaio. Poi posizionate la basetta di supporto superiore con la parte più lunga rivolta verso le ruote della Smart Car. Fissate le due viti posteriori da 3x8mm. Posizionate le due flange di sostegno del

sensore a infrarossi e fissatele alla basetta e alle colonnine anteriori con due viti 3x8mm.

5. Collegate il cavo al sensore ad ultrasuoni;
6. Fate passare il cavo nel foro rettangolare della basetta di supporto.
Collegatelo al connettore sulla scheda di espansione ed infine fissate il sensore ad ultrasuoni alle due flange con le viti da 3x8mm e relativi dadi. Viti inserite dalla parte anteriore e dadi dalla parte delle flange;
7. Verificate che il deviatore di alimentazione sia in posizione "spento", ovvero con la levetta verso il connettore della batteria. Posizionate la batteria sul suo alloggiamento e collegate i cavi al corrispondente connettore, di fianco al deviatore di accensione, facendo attenzione al verso. Collegate i cavi dei motori agli appositi connettori a fianco degli stessi facendo attenzione, anche in

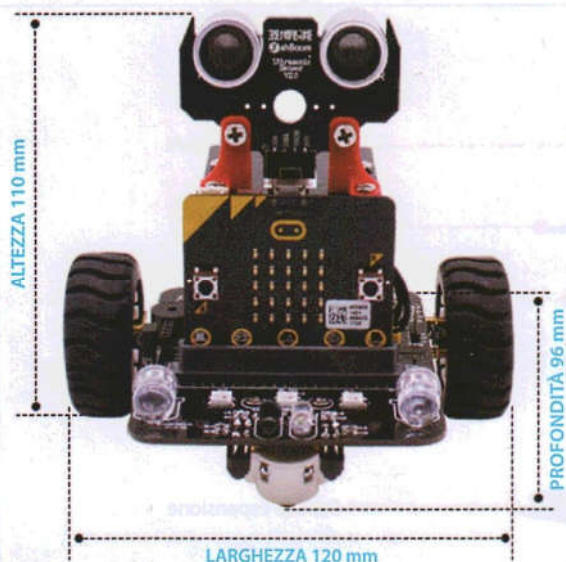


Fig. 5 - Smart Car completamente assemblata e relativi ingombri.

- questo caso, al verso di collegamento;
8. Inserite il micro:bit nell'apposito connettore, con i LED rossi rivolti verso la parte anteriore della Smart Car, quella con il ruotino. Le ruote di trazione sono nella parte posteriore.

In Fig. 5 è visibile l'aspetto finale della Smart Car con le relative dimensioni. Una volta completato l'assemblaggio, come prima operazione è oppor-

tuno eseguire una ricarica della batteria. Collegate un alimentatore da 5V con connettore micro USB (quello dei telefonini), al connettore "CHARGE" della scheda di supporto, con il deviatore nella posizione di spento. Si accenderà il LED rosso che indica che la batteria è in carica. La fine della carica verrà indicata dallo spegnimento del LED rosso e dall'accensione di quello verde di fine carica. Ora siamo pronti per la prima programmazione della Smart Car.

PROGRAMMAZIONE ED UTILIZZO

Come già anticipato nell'articolo citato precedentemente, sono disponibili diversi ambienti per la programmazione del micro:bit, la caratteristica della maggior parte di questi è la loro disponibilità via web, senza necessità di installare software sul proprio PC. Dopo aver caricato nella cache del browser le applet degli ambienti di sviluppo, è possibile scrivere i propri programmi anche in assenza di un collegamento internet. Per i dispositivi mobili, come tablet e smartphone, oltre che gli ambienti web sopra citati sono disponibili anche "app" gratuite scaricabili dagli store dei provider.

Prima di iniziare dobbiamo collegare il micro:bit al nostro PC utilizzando il cavo USB incluso nella confezione. Una volta collegato, micro:bit dovrebbe

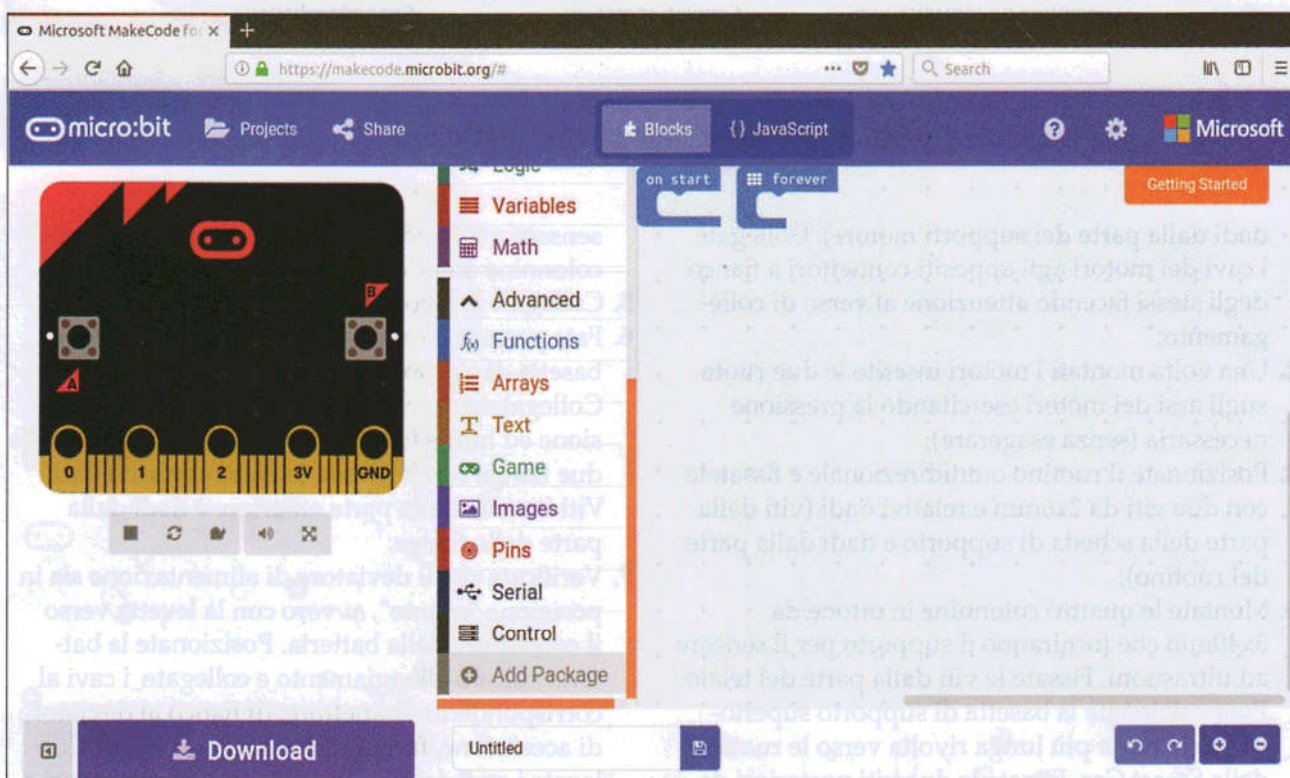


Fig. 6 - Ambiente di sviluppo Javascript Block Editor.

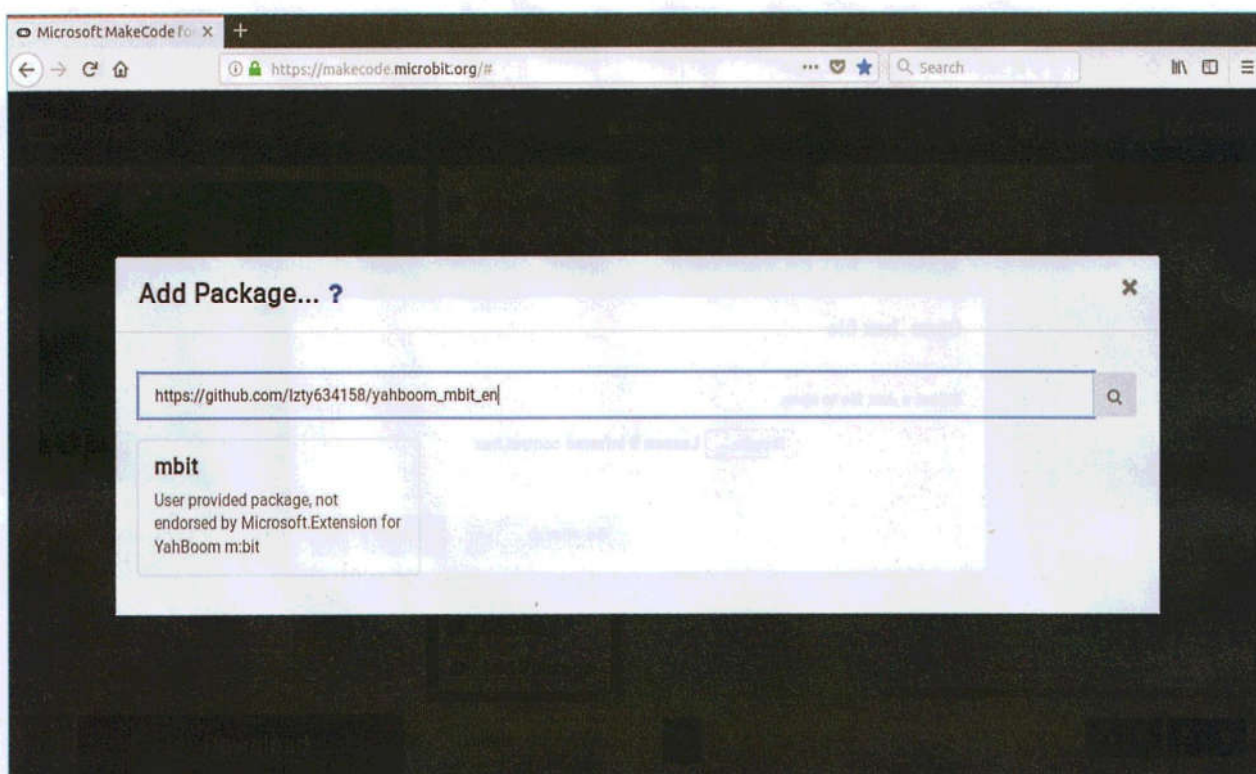


Fig. 7 - Finestra per aggiungere i blocchi di controllo della Smart Car.

venire riconosciuto immediatamente come una memoria di massa USB. Nel caso di Windows XP verrà richiesto di installare il relativo driver. Ora collegatevi al sito di riferimento di micro:bit all'indirizzo:

<http://microbit.org/>

Nella pagina principale cliccate sul pulsante "Let's Code" e vi troverete nella pagina che elenca i diver-

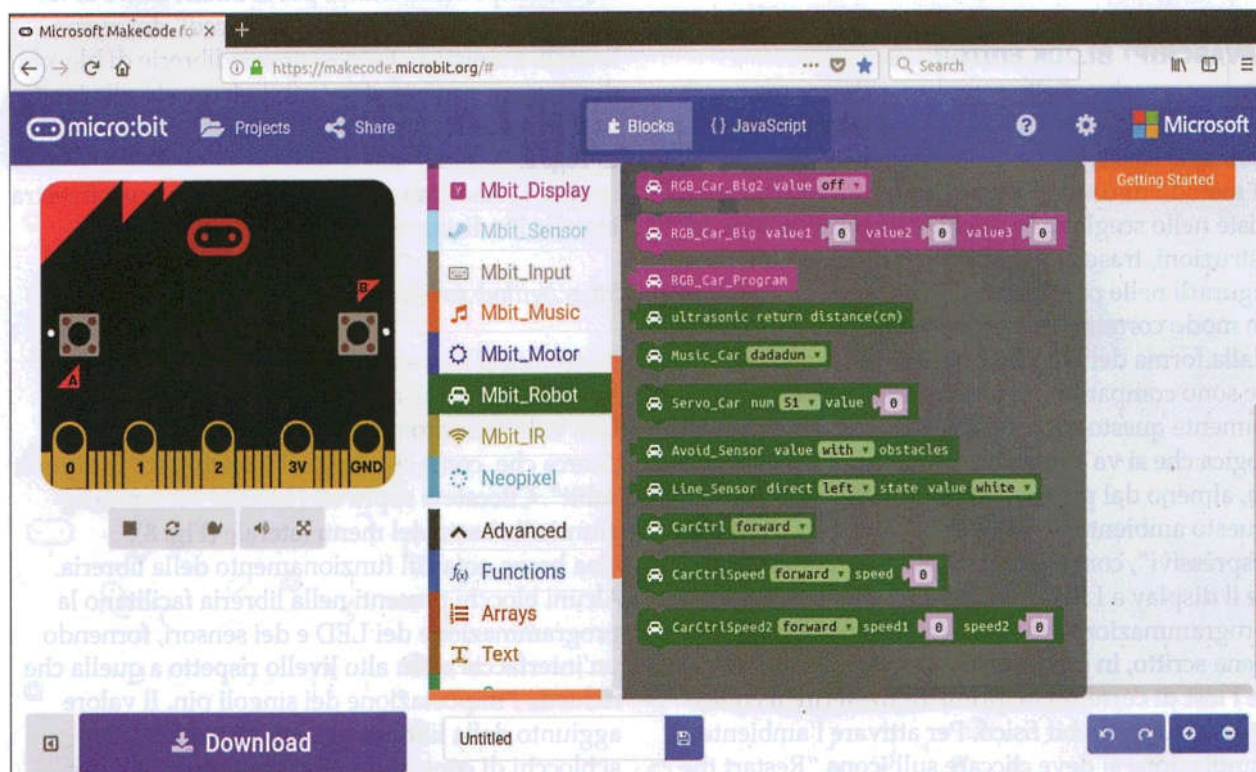


Fig. 8 - Nuovi blocchi disponibili per il controllo della Smart Car.

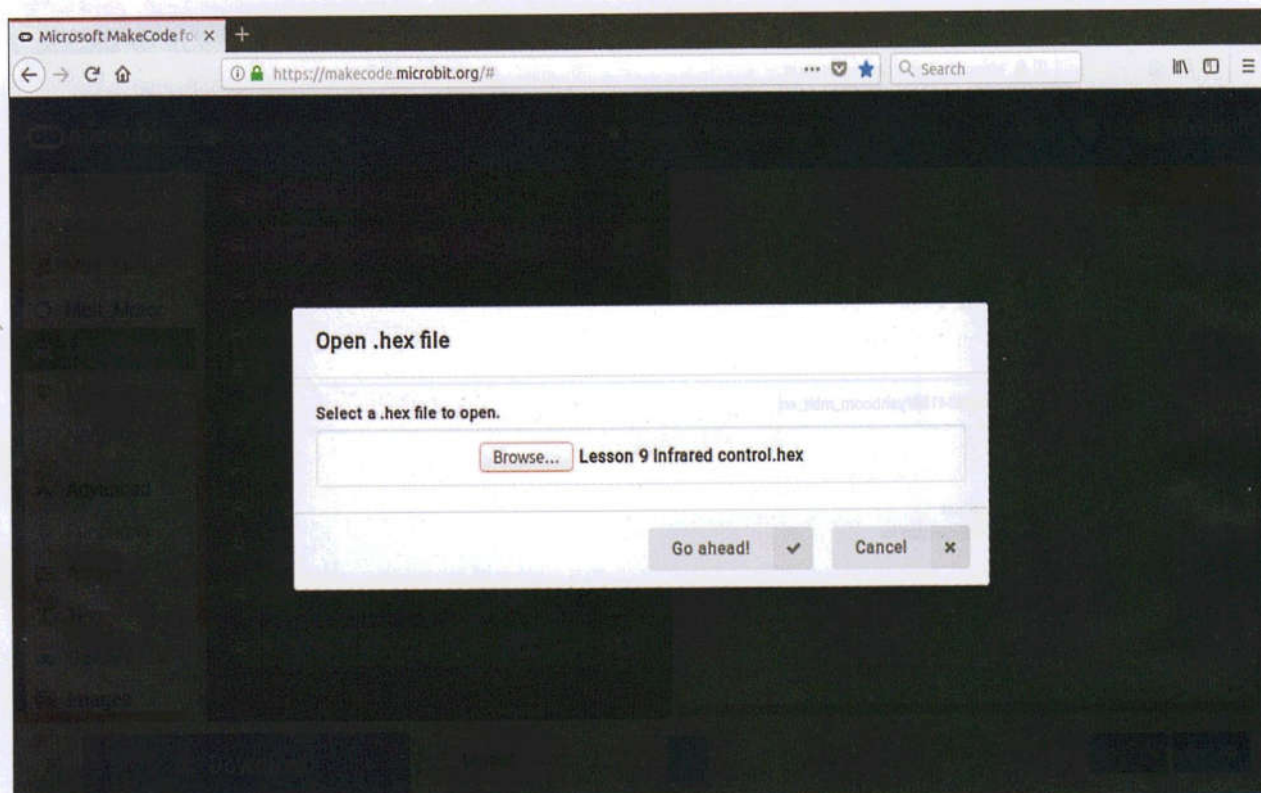


Fig. 9 - Selezione del programma da caricare nell'ambiente di sviluppo.

si ambienti di programmazione. Per programmare il micro:bit in modo da poter interagire con la scheda di espansione della Smart Car dobbiamo utilizzare l'ambiente di sviluppo Javascript Block Editor, un ambiente di sviluppo a blocchi che si ispira alla soluzione adottata da Scratch.

JAVASCRIPT BLOCK EDITOR

Come Scratch è molto semplice da usare ma non è un giocattolo. Permette di approfondire la programmazione strutturata e la gestione di eventi.

Il metodo di composizione di un programma consiste nello scegliere i blocchi che rappresentano le istruzioni, trascinarli nel pannello dell'editor, configurarli nelle parti personalizzabili ed "incastarli" in modo corretto. In questa operazione si è facilitati dalla forma dei blocchi. I blocchi si connettono solo se sono compatibili e le forme corrispondono. Chiamamente questo meccanismo non garantisce che la logica che si va a realizzare sia formalmente corretta, almeno dal punto di vista sintattico. Anche in questo ambiente di sviluppo vi sono blocchi molto "espressivi", come quello disponibile per configurare il display a LED. Una volta terminata la fase di programmazione è possibile eseguire il codice, appena scritto, in ambiente di simulazione, per eseguire i test di correttezza, prima di trasferire il codice stesso sul micro:bit fisico. Per attivare l'ambiente di simulazione si deve cliccare sull'icona "Restart the simulation", quella centrale in basso a sinistra. Una

volta entrati nell'ambiente di simulazione possiamo provare le diverse funzionalità.

Dopo aver aperto l'ambiente di sviluppo, per iniziare a programmare la Smart Car, è necessario arricchire l'ambiente di sviluppo stesso con l'estensione al Javascript Block Editor per la Smart Car. Per far questo cliccate sulla voce "Advanced" del menu laterale a sinistra, che riassume le librerie di blocchi di programmazione disponibili. In fondo alla tendina cliccate sulla voce "Add Package" come visibile in Fig. 6.

Si apre la finestra visibile in Fig. 7. In questa finestra inserite l'indirizzo:

https://github.com/lzty634158/yahboom_mbit_en

che punta al repository GitHub che contiene la libreria di blocchi specializzata per la Smart Car. Una volta inserito l'indirizzo cliccate sull'icona di ricerca che, come risultato, vi visualizzerà la libreria "mbit". Cliccatevi sopra ed i blocchi verranno aggiunti alla barra del menu laterale (Fig. 8).

Una breve nota sul funzionamento della libreria. Alcuni blocchi presenti nella libreria facilitano la programmazione dei LED e dei sensori, fornendo un'interfaccia a più alto livello rispetto a quella che richiede l'impostazione dei singoli pin. Il valore aggiunto della libreria è quello di offrire diversi blocchi di controllo, come appunto quelli per l'impostazione dei LED e per il controllo dei motori,



Fig. 10 - Codice per il controllo della Smart Car con il telecomando.

che interfacciano i singoli pin di generazione PWM dell'integrato PCA9685. Questa impostazione permette di ampliare di molto le funzionalità disponibili nel micro:bit usato da solo e di semplificarne, nel contempo, al massimo, l'utilizzo.

Come primo esempio di utilizzo, abbiamo scelto un'applicazione, tra quelle disponibili sul sito di tutoring della Smart Car, che permette di comandare il funzionamento del "robotino" utilizzando il telecomando a infrarossi. Per scaricarla andate al sito http://www.yahboom.com/study_en/Bitbot_EN. Scorrete la pagina fino alla Sezione "4 - Robot lessons". Cliccate sulla voce "9 - Infrared Control". Nella pagina che si apre potete eseguire il download di una presentazione del progetto in Powerpoint ed il codice del progetto stesso. Scaricateli entrambi in una cartella del vostro PC. Nel file Powerpoint è visibile una breve presentazione dove potete trovare lo schema delle funzioni assegnate al telecomando. Ora, dalla finestra di programmazione, caricate il programma nell'ambiente di sviluppo, cliccando sull'icona a forma di cartelletta e scegliendo poi il programma "microbit-Lesson-9-Infrared-control.hex" del quale avete appena eseguito il download come visibile in Fig. 9. Cliccate sul pulsante "Go ahead" per continuare.

Data la quantità di documentazione disponibile per l'ambiente Scratch e, con riferimento all'articolo sul micro:bit citato in precedenza, ci limitiamo a suggerirvi solo alcune modifiche al programma che avete aperto per limitare la velocità della Smart Car.

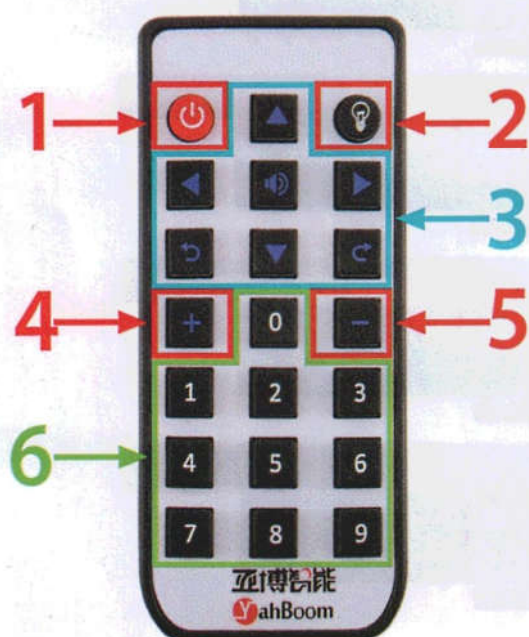


Fig. 11 - Il telecomando e le sue funzioni.

In tutti i blocchi dove vedete impostata la velocità a "255" modificate i valori come visibile in Fig. 10, in modo da limitare la velocità della Smart Car, il valore di velocità 255, che viene poi trasformato nel duty cycle assegnato al PWM di controllo dei motori, la rende un oggetto incontrollabile. Con l'occasione riordinate anche i blocchi in modo da renderne più ... razionale la disposizione. Un esempio è visibile in Fig. 10. Fatto?

Eseguite il download del programma, una volta in una vostra cartella in modo da avere un backup e poi, successivamente, nel file system del micro:bit, che dovrebbe apparirvi tra le memorie di massa del vostro PC con la sua letterina di identificazione. Ricordate di collegare il cavo USB proveniente dal vostro Pc al connettore USB del micro:bit e non a quello della scheda di espansione, altrimenti non si carica nulla.

Alla fine del download, scollegate il cavo USB e prendete il telecomando. La prima volta dovete togliere la protezione della batteria, altrimenti ... niente infrarossi. Una breve descrizione delle funzioni del telecomando è visibile in Fig. 11. Il tasto "1" spegne tutti i LED eventualmente accesi. Il tasto "2", col disegno della lampadina, accende i LED RGB con luce bianca. I tasti "4" ("+") e "5" ("-") permettono di accendere i LED con luce, rispettivamente verde e rossa. I tasti del blocco "2", permettono di pilotare il movimento dalla Smart Car. Il significato dei tasti è intuibile dai disegni di riferimento. I tasti del blocco "6", da zero a nove, invece, permettono di presentare sul display a matrice di LED diverse configurazioni di disegni. Accendete la Smart Car con il deviatore di alimentazione. Attendete la partenza del programma e ... buon divertimento. Ovviamente questa è solo la base sulla quale potete iniziare a sperimentare, modificando questo o gli altri programmi di esempio.

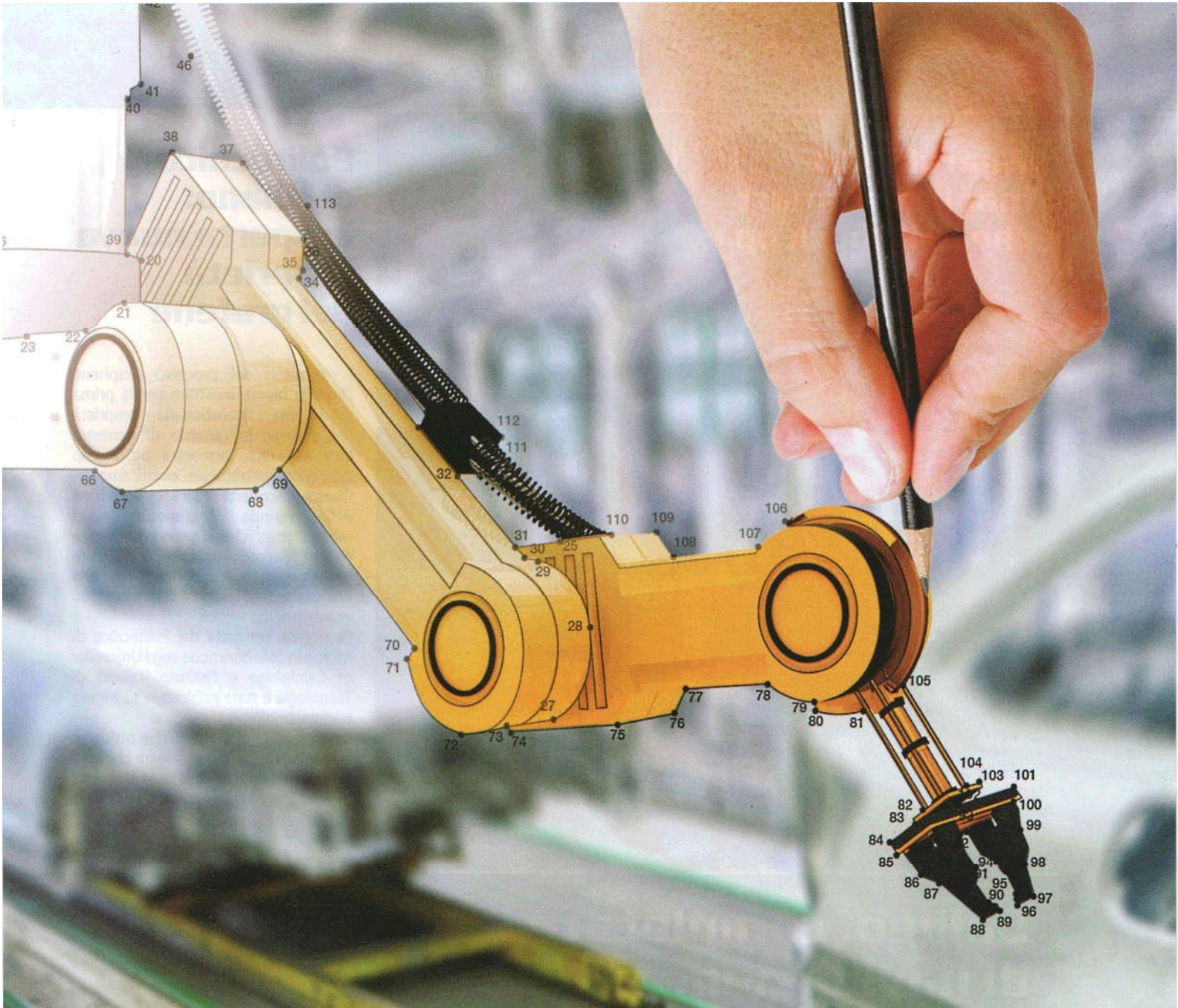


per il MATERIALE

Il kit per realizzare il robot con ruote basato su micro:bit (cod. MICROBITROBOT) è disponibile presso Futura Elettronica al prezzo di Euro 109,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287 - www.futurashop.it



Progetta senza sforzi con Microchip

Intelligenza Embedded di facile utilizzo supportata da Strumenti e Software di qualità di livello mondiale

www.microchip.com/effortless

microchip
DIRECT
www.microchipdirect.com



Il nome e logo Microchip ed il logo Microchip sono marchi industriali registrati di Microchip Technology Incorporated negli U.S.A. e in altri Stati. Tutti gli altri marchi industriali menzionati appartengono ai rispettivi titolari.
© 2018 Microchip Technology Inc. Tutti i diritti riservati. DS30010184A. MEC22211ta08/18

Wearable: la maglietta smart che fa il check up

Sviluppata da ENEA e Università Campus Bio-Medico di Roma, la maglietta "intelligente" acquisisce frequenza cardiaca e dati respiratori, grazie a una nuova applicazione tecnologica. Realizzata in tessuto elasticizzato con sensori in fibra ottica, faciliterà sia le indagini cliniche che la valutazione delle prestazioni sportive.

"Abbiamo utilizzato sensori in fibra ottica già in commercio e li abbiamo incapsulati all'interno di particolari materiali polimerici", spiega il ricercatore ENEA Michele Caponero. "Questi sensori, incorporati nelle magliette, risultano particolarmente utili in medicina sportiva, in quanto lasciano il paziente molto più libero nel movimento di quanto non avvenga con i metodi tradizionali, che prevedono l'utilizzo di un numero elevato di cinghie sul corpo", aggiunge Caponero. Queste magliette offrono vantaggi significativi anche per le indagini cliniche; ad esempio durante le risonanze magnetiche consentono di monitorare alcuni parametri legati agli atti respiratori, laddove i sensori tradizionali di tipo elettrico risultano



disturbati dall'intenso campo magnetico necessario per eseguire l'esame, senza nulla togliere al comfort del paziente, che deve limitarsi a indossare una maglia. Nella realizzazione della maglietta smart, ENEA si è occupata dell'aspetto tecnologico del sensore e del cablaggio, mentre l'Università Campus Bio-Medico ha ottimizzato il posizionamento dei sensori per misurare al meglio i parametri fisiologici.

www.enea.it

Politecnico di Milano: scoperta nuova proprietà del grafene

I ricercatori del progetto Graphene Flagship hanno mostrato per la prima volta una rivoluzionaria proprietà del grafene: la capacità di generare luce alla terza armonica ottica con efficienza controllabile elettricamente. Questa scoperta sarà alla base dello sviluppo di dispositivi ottici miniaturizzati che sfruttino frequenze ottiche precedentemente inutilizzate per trasmettere su banda larga una impressionante quantità di dati in modo estremamente veloce.

La ricerca, condotta dal Politecnico di Milano in collaborazione con l'University of Cambridge e l'Istituto Italiano di Tecnologia è stata pubblicata su Nature Nanotechnology.

La generazione di armoniche ottiche è un processo ottico non lineare che crea nuovi colori quando luce laser di elevata intensità interagisce con un materiale. In particolare, la generazione di terza ar-

Galileo, completato il sistema di navigazione europeo

Col lancio di altri quattro satelliti avvenuto il 25 luglio 2018, la rete di localizzazione satellitare europea Galileo raggiunge la piena operatività, portando al successo un progetto nato oltre 20 anni fa. Gli ultimi quattro satelliti sono stati lanciati a bordo di un razzo Ariane 5 dallo spazioporto europeo di Kourou; circa nove minuti dopo il decollo è avvenuta la separazione dello stadio superiore del razzo: la prima coppia di satelliti è stata rilasciata circa 3 ore dopo, seguita dalla seconda, 20 minuti più tardi. Il quarto si è poi posizionato

all'altitudine prevista, circa 22.922 km. Nei prossimi mesi i satelliti verranno posti nelle loro orbite definitive per poi iniziare un periodo di test della durata di sei mesi, che servirà a testare le loro funzionalità prima che possano unirsi al resto della co-

stellazio-
ne. Il lancio
- il numero 99 per
Ariane 5 - porta la co-
stellazione da 22 a 26 satelliti,
necessari per la piena operatività
del sistema, migliorando ulteriormente



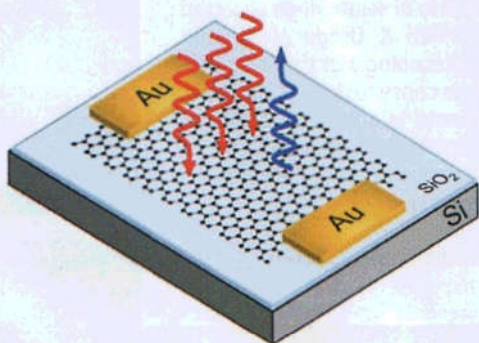
la precisione e la copertura globale. I Servizi Iniziali del sistema Galileo sono cominciati il 15 dicembre 2016, e oggi, più di 100 milioni di dispositivi commerciali utilizzano Galileo.

www.esa.int

monica produce luce la cui frequenza è il triplo di quella della luce incidente, quindi a partire dalla luce infrarossa invisibile può produrre una intensa luce visibile. Il grafene, nonostante sia il materiale più sottile esistente perché ha lo spessore di un solo strato di atomi, ha una risposta ottica non lineare sorprendentemente elevata.

I partner del progetto Graphene Flagship hanno dimostrato per la prima volta che il processo di generazione di terza armonica in grafene può essere controllato in maniera molto efficace mediante l'applicazione di un campo elettrico esterno. In altre parole, la luce di colore visibile generata dal grafene può essere accesa o spenta in modo molto semplice tramite una tensione applicata al materiale.

www.graphene-flagship.eu



Pronto per il primo volo il BelugaXL di nuova generazione

La nuova generazione di aerei da trasporto Beluga di Airbus, contraddistinta dal suffisso XL, è pronta per il volo inaugurale. Per questa occasione, è stata scelta una livrea particolare, frutto di una votazione tra i dipendenti Airbus: una balena sorridente.

Con una fusoliera superiore ed una enorme area di carico, Airbus BelugaXL è ancora più grande del suo predecessore Beluga A300-600ST essendo 6 metri più lungo, un metro più largo e con una capacità di carico superiore di 4 tonnellate. La flotta di cinque Beluga XL garantirà nei prossimi anni il trasporto di parti anche molto grandi degli aerei in costruzione dai

siti produttivi dell'azienda in tutta Europa e verso le linee di assemblaggio finali in Francia, Germania e Spagna.

La verniciatura conclude una fase di collaudo a terra completa, comprendente un totale di 72 valutazioni specifiche richieste prima del volo inaugurale tra cui un test su terreno pesante e una prova di resistenza alle vibrazioni durata otto giorni.

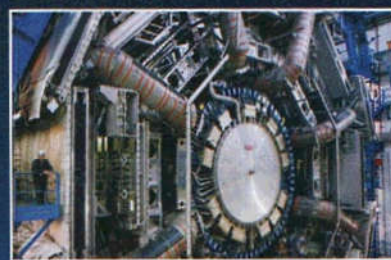
Dopo il volo inaugurale inizierà la campagna di certificazione dei test di volo della durata di 10 mesi che porterà all'ingresso in servizio previsto per il 2019.

www.airbus.com

Osservato l'Higgs che produce bellezza

Si tratta di un processo, previsto dal Modello Standard delle particelle, cercato a lungo: ci sono voluti, infatti, sei anni per riuscire a identificare il decadimento del bosone di Higgs in una coppia di particelle fondamentali, chiamate quark b (da beauty ovvero bellezza). L'osservazione, che è stata presentata il 28 agosto, al CERN, dalle collaborazioni scientifiche degli esperimenti ATLAS e CMS all'acceleratore LHC, conferma l'ipotesi secondo cui il campo quantico del bosone di Higgs, che pervade tutto l'universo, si accoppi anche al quark b fornendogli massa. Il Modello Standard della fisica delle particelle prevede che circa il 60% delle volte il bosone di Higgs decada in una coppia di quark b: trovarne conferma sperimentale era

importante per verificare anche questa previsione della nostra teoria. Ma individuare questo canale di decadimento è stato tutt'altro che facile, come dimostrato dal lungo periodo trascorso dalla scoperta del famoso bosone prima di riuscire a mettere a segno anche questo risultato. La ragione della difficoltà risiede nel fatto che ci sono molti altri modi per produrre quark b nelle collisioni tra protoni. Ciò rende difficile isolare il segnale di decadimento del bosone di Higgs dal "rumore di fondo" associato a questi processi. Al contrario, i canali di decadimento del bosone di Higgs meno comuni, che sono stati osservati al momento della scoperta della particella, come il decadimento in una coppia di fotoni, sono molto più facili da identifi-



care ed estrarre dal fondo. Per isolare in modo definitivo il segnale, le collaborazioni ATLAS e CMS hanno dovuto combinare i dati del primo e del secondo ciclo di attività (run) di LHC, che riguardavano le collisioni alle energie di 7, 8 e 13 TeV. Il risultato, sia per ATLAS sia per CMS, è stato l'osservazione del decadimento del bosone di Higgs in una coppia di quark b con una significatività statistica superiore a 5 deviazioni standard (5 sigma).

www.infn.it

Leonardo Company e Polimi per l'elicottero del futuro

Leonardo e il Politecnico di Milano ampliano la collaborazione, avviata nel 2016, con la firma di Innovation Hub, un accordo quadro di valenza pluriennale. L'intesa, oltre a soluzioni innovative relative agli elicotteri, riguarda ulteriori ambiti di ricerca che attengono al business di Leonardo. La partnership tra l'Azienda e l'ateneo milanese riafferma l'importanza della collaborazione tra grandi imprese e università di eccellenza come fattore determinante di sviluppo e crescita economica e culturale.

La collaborazione tra Politecnico di Milano e Leonardo è volta alla realizzazione di progetti di innovazione concreti: oggetto della ricerca saranno i componenti innovativi di accoppiamento meccanico basati su nuovi materiali, essenziali per consentire il trasferimento di carichi dal propulsore alla scatola di trasmissione principale dell'elicottero. Ciò consentirà una significativa semplificazione del sistema meccanico, grazie all'utilizzo di un singolo componente flessibile anziché di

un sistema articolato.

Attraverso il progetto Comfort, Leonardo e Politecnico di Milano stanno sviluppando, inoltre, tecnologie utili per la riduzione delle vibrazioni e del rumore degli elicotteri. La piattaforma di riferimento per lo studio è rappresentata dall'AW139. Le attività di ricerca si concentrano su tre aree principali: il rotore, da cui provengono le vibrazioni, i sistemi di vincolo della trasmissione, attraverso cui tali vibrazioni sono diffuse alla struttura dell'elicottero riducendo il livello di comfort e infine alla cabina stessa. L'obiettivo

finale del programma è dimostrare un miglioramento significativo nella riduzione della vibrazione e del rumore interno, attraverso un uso coordinato di sistemi passivi e attivi. Prenderà il via entro l'anno, nell'ambito dell'accordo, un innovativo progetto di manutenzione predittiva dello stato di salute degli elicotteri HUMS (Health & Usage Monitoring Systems), tecnologia in continuo progresso, che consentirà di indicare preventivamente le anomalie di funzionamento permettendo un incremento dei livelli di sicurezza e di riduzione dei costi di esercizio.

www.leonardocompany.com



Ai blocchi di partenza Ital-GovSatCom

Il Programma Mirror GovSatCom ha come obiettivo la realizzazione e attivazione di un sistema satellitare innovativo, denominato Ital-GovSatCom, per l'erogazione di servizi di telecomunicazioni con caratteristiche di sicurezza, resilienza ed affidabilità tali da consentirne l'utilizzo per finalità istituzionali. Tra i campi di applicazione, protezione civile, sicurezza, difesa, aiuto umanitario, telemedicina, sorveglianza marittima e altro ancora. Entra così in fase di attuazione il Piano Strategico Nazionale Space Economy che prevede un investimento di circa 4,7 miliardi di euro. Si tratta del programma nazionale mirror della nascente iniziativa europea denominata, appunto, GovSatCom, tale da permettere all'Italia un importante posizionamen-

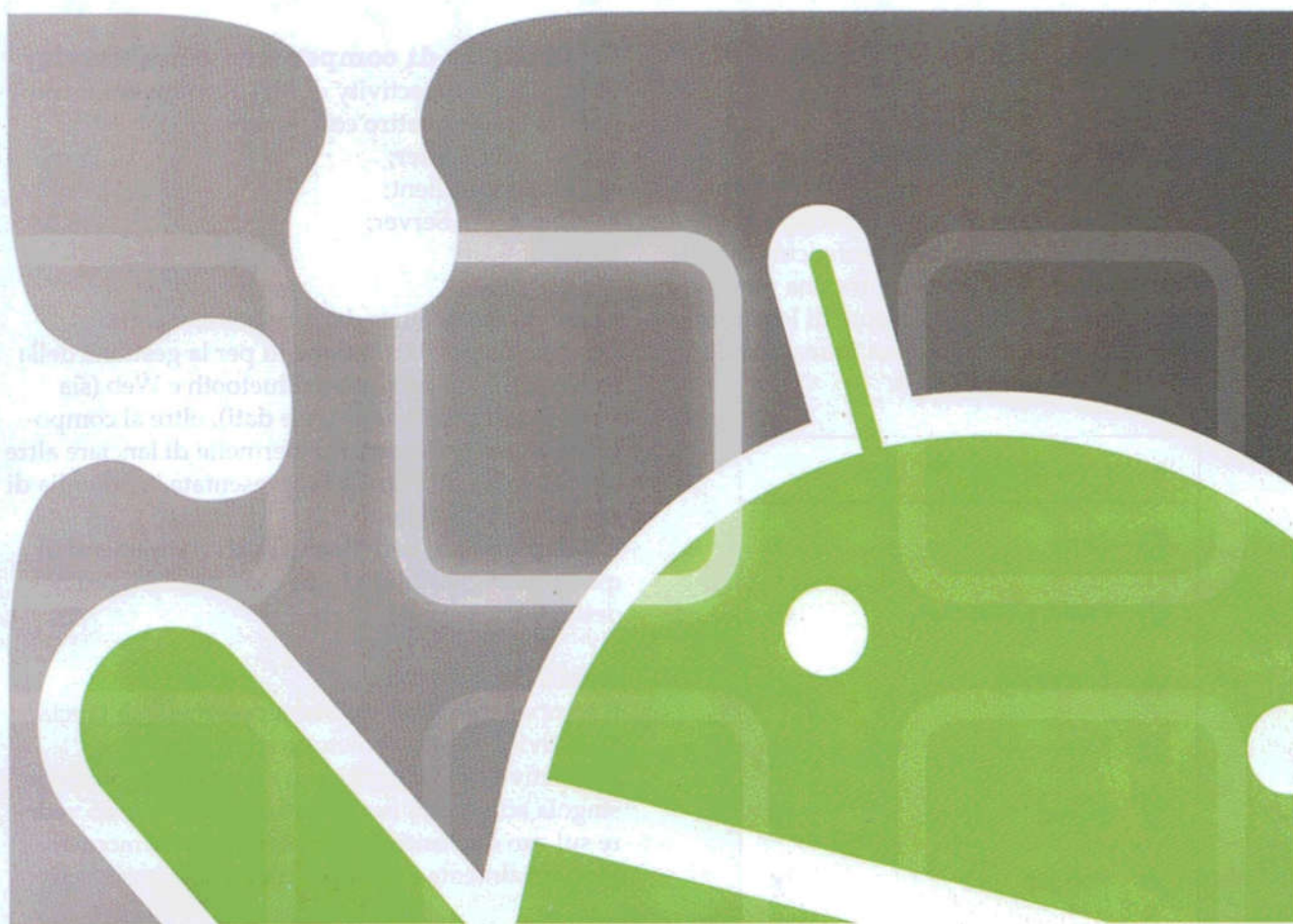
to in un settore strategico come quello delle telecomunicazioni istituzionali. Il Programma Mirror GovSatCom è realizzato sulla base di un accordo di collaborazione istituzionale sottoscritto tra MISE ed Agenzia Spaziale Italiana, e di un Protocollo d'Intesa sottoscritto dal MISE e da 12 Regioni (Abruzzo, Basilicata, Campania, Emilia Romagna, Lazio, Lombardia, Piemonte, Puglia, Sardegna, Toscana, Umbria, Valle D'Aosta) e dalla Provincia autonoma di Trento.

Per garantire la competitività dei ser-

vizi offerti, il sistema Ital-GovSatCom sarà basato su un'innovativa piattaforma geostazionaria con propulsione elettrica e caratteristiche di peso e dimensione tali da garantirne la lancia-bilità a basso costo, con vettori di tipo Vega.

Lo sviluppo di una piattaforma di piccola taglia, oltre a soddisfare i requisiti posti per il sistema Ital-GovSatCom, darà al nostro Paese la possibilità di inserirsi con un proprio autonomo ruolo nella realizzazione di satelliti modulari e multi uso, competitivi sia in termini di prezzo che di caratteristiche tecniche.

www.asi.it



MIT APP INVENTOR

6

di FRANCESCO FICILI

Continuiamo il nostro viaggio alla scoperta di MIT App Inventor, un tool di sviluppo per applicazioni Android creato dal MIT. In questa sesta e conclusiva puntata analizzeremo le famiglie di componenti maps e connectivity e vedremo come pubblicare le nostre app su google play.

Nella puntata precedente abbiamo proseguito lo studio e l'applicazione pratica dei componenti di MIT Appinventor, analizzando la famiglia di componenti **drawing and animations**, che ci permette di sviluppare applicazioni di gaming sui nostri dispositivi Android. Ricordiamo che i componenti sono gli elementi che permettono di aggiungere funzionalità all'app in fase di sviluppo con App Inventor. Ebbene, in questa puntata concludiamo il discorso e presentiamo due ulteriori famiglie di componenti, ossia **maps** e **connectivity**, e vediamo anche come

fare per pubblicare, sullo store Google Play, le app che abbiamo creato.

La famiglia di componenti Maps

App Inventor dispone della famiglia di componenti Maps, appositamente creati per lo sviluppo di applicazioni di geolocalizzazione e geofencing (applicazioni che sfruttano l'interazione tra una mappa e le informazioni ricavate dal sensore di localizzazione). In Fig. 1 è rappresentata un'immagine dei componenti della famiglia Maps.

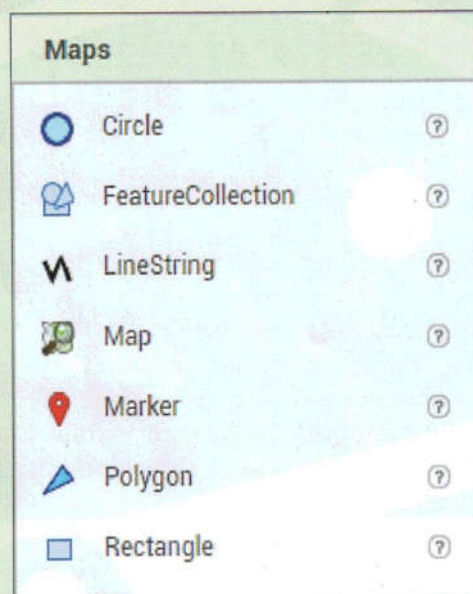


Fig. 1 - Famiglia di componenti Maps.

Non ci soffermeremo molto sulla descrizione di questi componenti in quanto non li utilizzeremo nell'esempio pratico presentato in questa puntata; ci limiteremo piuttosto a fornire una tabella riassuntiva che ne illustra le caratteristiche generali.

COMPONENTE	DESCRIZIONE
Circle	Il circle component permette di visualizzare un cerchio di un dato raggio in metri e una certa latitudine e longitudine, all'interno di una mappa. Questo componente può essere utilizzato per l'implementazione di applicazioni di geofencing, utilizzando i metodi di interazione con il sensore di localizzazione.
FeatureCollection	Il FeatureCollection component raggruppa le feature di una o più mappe insieme.
LineString	Il LineString component permette di disegnare una sequenza continua di linee su una mappa.
Map	Il Map component è un container bidimensionale che permette di renderizzare mappe e piazzare vari tipi di marker al suo interno per identificare punti o zone. Le mappe sono fornite dal servizio OpenStreetMaps.
Marker	Il componente Marker permette di indicare punti in una mappa, come edifici o altri punti di interesse. I marker possono essere customizzati in diversi modi, come ad esempio cambiandone l'aspetto grafico utilizzando un asset grafico dell'app.
Polygon	Il componente Polygon permette di racchiudere un'area di dimensioni arbitrarie su di una mappa. L'uso principale di questo componente è quello di delimitare zone geografiche.
Rectangle	Il componente Rectangle permette di delimitare zone all'interno di un poligono rettangolare, con valori fissi per latitudine e longitudine dei vertici.

Tabella 1 - Componenti della famiglia Maps.

La famiglia di componenti connectivity

La famiglia connectivity di MIT App Inventor comprende questi quattro componenti:

- Activity Starter;
- Bluetooth Client;
- Bluetooth Server;
- Web.

Come potete dedurre dai loro nomi, si tratta principalmente di componenti per la gestione della connettività, in particolare Bluetooth e Web (sia tramite WiFi che tramite rete dati), oltre al componente Activity Starter, che permette di lanciare altre applicazioni. In Fig. 2 è rappresentata la famiglia di componenti in oggetto. Analizziamo più in dettaglio i vari componenti di questa famiglia, in modo da comprenderli più nel dettaglio.

Activity Starter

Il componente Activity Starter permette di lanciare un activity tramite il metodo start activity.

Una activity in Android può essere vista come una singola schermata (screen) che un utente può vedere sul suo dispositivo in un determinato momento. Sostanzialmente una app è composta da più attivi-

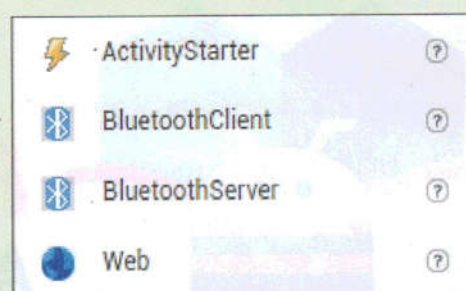


Fig. 2 - Famiglia di componenti connectivity.

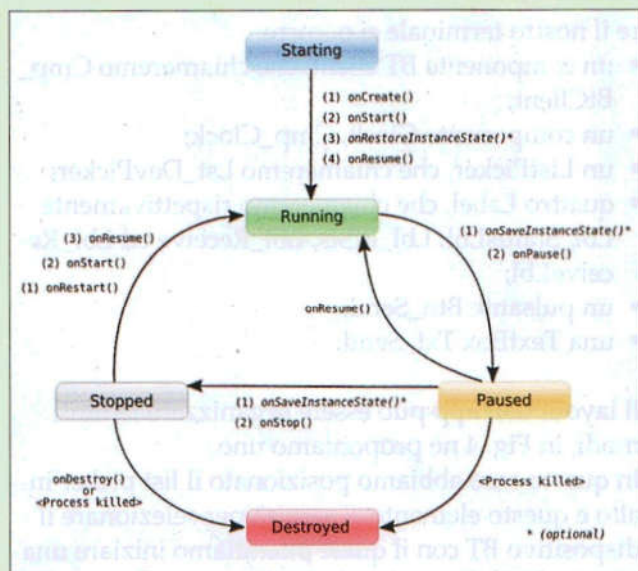


Fig. 3 - Ciclo di vita di una activity Android.

ty, ognuna delle quali può essere vista ed invocata da un'altra app, ed è a questa funzione che il componente **Activity Starter** assolve.

Una volta invocata, una activity ha anche un ciclo di vita, con tutta una serie di eventi correlati.

Il ciclo di vita delle activity in Android è riportato in Fig. 3.

Nella Tabella 2 sono riportati, come di consueto, eventi, metodi e proprietà del componente.

Tra le activity che possono essere lanciate dal componente activity starter possiamo citare, giusto per fare qualche esempio.

- Avviare l'applicazione camera, cosa che può esse-

re fatto impostando:

- Action: `android.intent.action.MAIN`
- ActivityPackage: `com.android.camera`
- ActivityClass: `com.android.camera.Camera`

- Eseguire una ricerca su web. Assumendo di voler cercare "ElettronicaIn", bisognerebbe impostare:

- Action: `android.intent.action.WEB_SEARCH`
- ExtraKey: `query`
- ExtraValue: `ElettronicaIn`
- ActivityPackage: `com.google.android.providers.enhancedgooglesearch`
- ActivityClass: `com.google.android.providers.enhancedgooglesearch.Launcher`

- Aprire il browser web ad una specifica pagina. Assumendo di voler aprire la pagina web di Elettronica In, bisognerebbe impostare:

- Action: `android.intent.action.VIEW`
- DataUri: `https://www.elettronica.in.it/`

Bluetooth Client/Server

I moderni smartphone basati su Android sono normalmente dotati di un modulo bluetooth integrato. App Inventor modella il collegamento bluetooth tramite uno schema client/server; esistono quindi due componenti per la gestione del BT, a seconda che si voglia implementare il lato client o il lato server.

Il profilo BT maggiormente supportato è il profilo SPP (Serial Port Profile) che è il profilo base utilizzato dalla maggior parte dei metodi. C'è comunque la possibilità di connettersi tramite un profilo generico, ma il supporto specifico a livello di metodi è molto

PROPRIETÀ

SINTASSI	DESCRIZIONE
Action	Action della activity.
Activity Class	Class della activity.
Activity Package	Package della activity.
Data Type	DataType della activity.
Data Uri	Data Uri della activity.
Extras	Questa proprietà accetta una lista di coppie key/value che possono poi essere utilizzate nel campo Extras della activity stessa. Sostituisce le proprietà obsolete ExtraKey ed ExtraValue.

EVENTI

SINTASSI	DESCRIZIONE
AfterActivity(text result)	Evento triggerato quando il metodo StartActivity ritorna.
ActivityCanceled()	Evento triggerato quando il metodo StartActivity ritorna prematuramente, nel caso l'activity sia stata cancellata.

METODI

SINTASSI	DESCRIZIONE
text ResolveActivity()	Ritorna il nome dell'activity corrispondente all'activityStarter, oppure una stringa vuota se l'activity corrispondente non viene trovata.
StartActivity()	Lancia l'activity corrispondente all'activityStarter.

Tabella 2 - Eventi, metodi e proprietà del componente Activity Starter.

limitato in questo caso.

Nella **Tabella 3** sono riportati eventi, metodi e proprietà del componente BT client.

Invece nella **Tabella 4** sono riportati eventi, metodi e proprietà del componente BT Server.

Web

Il componente **Web** è un elemento di App Inventor non visibile che permette di gestire richieste http GET, POST, PUT e DELETE. Il componente prescinde dal tipo di connessione fisica utilizzata (WiFi o GSM/GPRS), fornendo un'interfaccia uniforme per l'accesso alla rete.

Nella **Tabella 5** sono riportati, come di consueto, eventi, metodi e proprietà del componente.

Esempio Pratico

Passiamo adesso, come di consueto, al nostro esempio pratico, per il quale utilizzeremo il componente BT, creando un semplice terminale BT con profilo SPP (Serial Port Profile).

Come di consueto partiamo dal layout; per realizza-

re il nostro terminale ci occorre:

- un componente BT client, che chiameremo Cmp_BtClient;
- un componente Clock, Cmp_Clock;
- un ListPicker, che chiameremo Lst_DevPicker;
- quattro Label, che chiameremo rispettivamente Lbl_StatusLbl, Lbl_BtSts, Lbl_Receive ed Lbl_ReceiveLbl;
- un pulsante Btn_Send;
- una TextBox Txt_Send.

Il layout dell'app può essere organizzato in tanti modi, in Fig. 4 ne proponiamo uno.

In questo caso abbiamo posizionato il list picker in alto e questo elemento ci servirà per selezionare il dispositivo BT con il quale intendiamo iniziare una connessione. Sotto al picker abbiamo inserito una label che ci indicherà lo stato della connessione. Più in basso ancora la text box ed il pulsante per l'invio ed infine la label sulla quale stamperemo le stringhe in ricezione. Per quanto riguarda la ricezione configureremo il componente clock per generare un

PROPRIETÀ	
SINTASSI	DESCRIZIONE
AddressesAndNames	Proprietà che contiene indirizzi e nomi dei dispositivi BT accoppiati.
Available	Proprietà che indica se il modulo BT è presente sul dispositivo.
Enabled	Proprietà che indica se il modulo BT è abilitato.
IsConnected	Proprietà che indica se è stata stabilita una connessione.
DelimiterByte	Proprietà che permette di settare il delimiterByte (byte di delimitazione).
EVENTI	
SINTASSI	DESCRIZIONE
Nessuno	
METODI	
SINTASSI	DESCRIZIONE
boolean Connect(text address)	Metodo che permette di effettuare una connessione con il dispositivo il cui indirizzo è passato come parametro, mediante profilo SPP. Ritorna un boolean che indica se la connessione è andata a buon fine.
boolean ConnectWithUUID(text address, text uuid)	Metodo che permette di effettuare una connessione con il dispositivo il cui indirizzo è passato come parametro, mediante un profilo generico (UUID). Ritorna un boolean che indica se la connessione è andata a buon fine.
Disconnect()	Metodo che permette di disconnettersi dal dispositivo BT con il quale si è connessi.
boolean IsDevicePaired(text address)	Metodo che indica se un determinato dispositivo, il cui indirizzo è passato come parametro, è accoppiato.
text ReceiveText(number numberOfBytes)	Metodo che ritorna una stringa di testo ricevuta dal dispositivo connesso, di lunghezza pari al parametro NumberOfBytes. Se il parametro NumberOfBytes è minore di 0 vengono letti tutti i bytes ricevuti prima del carattere di delimitazione.
list ReceiveSignedBytes(number numberOfBytes)	Metodo che ritorna lista di bytes con segno ricevuta dal dispositivo connesso, di lunghezza pari al parametro NumberOfBytes. Se il parametro NumberOfBytes è minore di 0 vengono letti tutti i bytes ricevuti prima del carattere di delimitazione.
list ReceiveUnsignedBytes(number numberOfBytes)	Metodo che ritorna lista di bytes senza segno ricevuta dal dispositivo connesso, di lunghezza pari al parametro NumberOfBytes. Se il parametro NumberOfBytes è minore di 0 vengono letti tutti i bytes ricevuti prima del carattere di delimitazione.
SendBytes(list list)	Metodo che consente di inviare una lista di bytes al dispositivo BT connesso.
SendText(text text)	Metodo che consente di inviare una stringa di caratteri al dispositivo BT connesso.

Tabella 3 - Eventi, metodi e proprietà del componente BT Client.

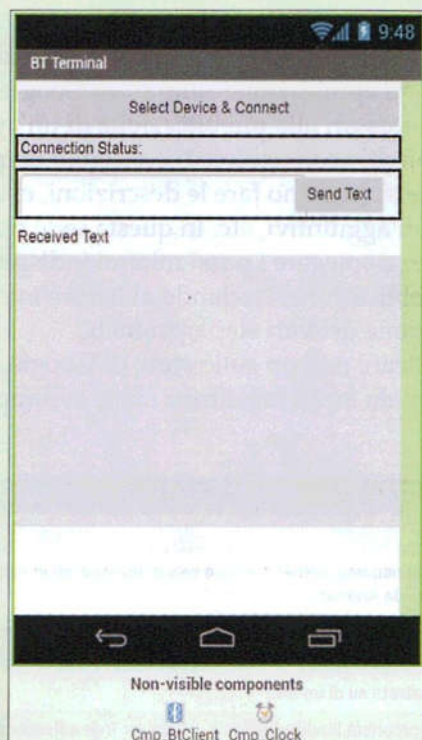


Fig. 4 - Layout dell'applicazione di esempio.

evento ogni 200ms e all'interno dell'evento accoderemo le stringhe ricevute dal BT se ce ne sono. Analizziamo adesso i behaviours, partendo dall'inizializzazione e dalla connessione con il server BT, operazioni rappresentate dai blocchi in Fig. 5. In corrispondenza dell'evento "Initialize" dello screen viene inizializzata la label Lbl_BtSts. Successivamente si popola la lista Lst_DevPicker con indirizzi e nomi dei dispositivi BT associati, prendendoli dalla proprietà AdresseAndNames del componente, in corrispondenza dell'evento "BeforePicking". La selezione ottenuta in questa fase (Lst_DevPicker.Selection) viene utilizzata in corrispondenza dell'evento "AfterPicking" per connettersi col dispositivo sfruttando il metodo "Connect" del componente BT Client. Sempre in questo evento viene aggiornato il valore della label Lbl_BtStatus a "Connected". La gestione dell'invio e della ricezione dei dati è molto semplice. Per inviare una stringa, in corrispondenza dell'evento "Btn_Send.Click" viene invocato il metodo "SendText" del componente BT Client (viene preventivamente verificato che il componente sia connesso), come illustrato in Fig. 6.

PROPRIETÀ	
SINTASSI	DESCRIZIONE
Available	Proprietà che indica se il modulo BT è presente sul dispositivo.
DelimiterByte	Proprietà che permette di settare il delimiterByte (byte di delimitazione).
Enabled	Proprietà che indica se il modulo BT è abilitato.
IsAccepting	Proprietà che indica se il componente BT server sta accettando una connessione
IsConnected	Proprietà che indica se è stata stabilita una connessione.
EVENTI	
SINTASSI	DESCRIZIONE
ConnectionAccepted()	Evento che segnala che la connessione BT è stata accettata.
METODI	
SINTASSI	DESCRIZIONE
AcceptConnection(text serviceName)	Accetta una connessione via BT tramite profilo SPP.
AcceptConnectionWithUUID(text serviceName, text uuid)	Accetta una connessione via BT tramite profilo generico.
Disconnect()	Metodo che permette di disconnettersi dal dispositivo BT con il quale si è connessi.
number BytesAvailableToReceive()	Metodo che ritorna il numero di bytes che possono essere ricevuti.
list ReceiveSignedBytes (number numberOfBytes)	Metodo che ritorna lista di bytes con segno ricevuta dal dispositivo connesso, di lunghezza pari al parametro NumberOfBytes. Se il parametro NumberOfBytes è minore di 0 vengono letti tutti i bytes ricevuti prima del carattere di delimitazione.
text ReceiveText (number numberOfBytes)	Metodo che ritorna una stringa di testo ricevuta dal dispositivo connesso, di lunghezza pari al parametro NumberOfBytes. Se il parametro NumberOfBytes è minore di 0 vengono letti tutti i bytes ricevuti prima del carattere di delimitazione.
list ReceiveUnsignedBytes (number numberOfBytes)	Metodo che ritorna lista di bytes senza segno ricevuta dal dispositivo connesso, di lunghezza pari al parametro NumberOfBytes. Se il parametro NumberOfBytes è minore di 0 vengono letti tutti i bytes ricevuti prima del carattere di delimitazione.
SendBytes(list list)	Metodo che consente di inviare una lista di bytes al dispositivo BT connesso.
SendText(text text)	Metodo che consente di inviare una stringa di caratteri al dispositivo BT connesso.
StopAccepting()	Metodo per interrompere l'accettazione di una connessione.

Tabella 4 - Eventi, metodi e proprietà del componente BT Server.

Per la ricezione invece si sfrutta il componente clock. Ad ogni occorrenza dell'evento "Time" viene aggiornata la label Lbl_Receive, accodando i caratteri ricevuti sul canale BT mediante l'applicazione del metodo "ReceiveText", come illustrato in Fig. 7. Anche in questo caso si verifica se il componente BT è connesso).

In Fig. 8 è visibile uno screenshot che rappresenta il funzionamento su un dispositivo reale: è stato utilizzato un HC-05 (disponibile sul sito www.futurashop.it/HC05) che effettua un loopback sulla linea seriale.

Pubblicare le app su google play

Passiamo adesso ad una breve guida alla pubblicazione di una applicazione sullo store Google Play. Gli step necessari alla pubblicazione di un'app sono diversi e molti sono opzionali a seconda di quanto dettagliate si vogliano fare le descrizioni, eventuali contenuti aggiuntivi, etc. In questa sede noi ci limiteremo a spiegare i passi minimi indispensabili per la pubblicazione, lasciando al lettore interessato l'esplorazione dei vari step opzionali.

Per pubblicare un'app sullo store di Google, il primo step da fare è registrarsi come sviluppatore,

PROPRIETÀ	
SINTASSI	DESCRIZIONE
AllowCookies	Proprietà che indica se i cookies provenienti da una risposta debbano essere salvati ed utilizzati in richieste successive. Il supporto ai cookies è disponibile solo da Android 2.3.
RequestHeaders	Header della richiesta, come lista di due elementi. Il primo elemento rappresenta il field name della richiesta, mentre il secondo rappresenta il valore.
SaveResponse	Proprietà che indica se la risposta debba essere salvata su di un file.
ResponseFileName	Il nome del file nel quale salvare la risposta. Se la proprietà SaveResponse è settata su True e ResponseFileName è vuoto, viene creato un nuovo file con un nome di default.
Url	Url della web request.
EVENTI	
SINTASSI	DESCRIZIONE
GotFile(text url, number responseCode, text responseType, text fileName)	Evento che indica che una richiesta è stata servita e la risposta è stata salvata su un file.
GotText(text url, number responseCode, text responseType, text responseContent)	Evento che indica che una richiesta è stata servita ed il contenuto della risposta viene passato come stringa.
METODI	
SINTASSI	DESCRIZIONE
text BuildRequestData(list list)	Metodo che converte una lista di coppie di elementi nome-valore in una stringa formattata correttamente per l'uso del metodo PostText.
ClearCookies()	Cancella tutti i cookies del web component.
Delete()	Esegue una richiesta http DELETE utilizzando la proprietà url e recupera la risposta. Se la proprietà SaveResponse è settata a true, la risposta verrà salvata nel file indicato e verrà triggerato l'evento GotFile, altrimenti verrà triggerato l'evento GotText.
Get()	Esegue una richiesta http GET utilizzando la proprietà url e recupera la risposta. Se la proprietà SaveResponse è settata a true, la risposta verrà salvata nel file indicato e verrà triggerato l'evento GotFile, altrimenti verrà triggerato l'evento GotText.
PostFile(text path)	Esegue una richiesta http POST utilizzando la proprietà url ed i dati provenienti dal file specificato come parametro. Se la proprietà SaveResponse è settata a true, la risposta verrà salvata nel file indicato e verrà triggerato l'evento GotFile, altrimenti verrà triggerato l'evento GotText.
PostText(text text)	Esegue una richiesta http POST utilizzando la proprietà url e la stringa dati passata come parametro. I caratteri della stringa dati saranno codificati mediante codifica UTF-8. Se la proprietà SaveResponse è settata a true, la risposta verrà salvata nel file indicato e verrà triggerato l'evento GotFile, altrimenti verrà triggerato l'evento GotText.
PutFile(text path)	Esegue una richiesta http PUT utilizzando la proprietà url ed i dati provenienti dal file specificato come parametro. Se la proprietà SaveResponse è settata a true, la risposta verrà salvata nel file indicato e verrà triggerato l'evento GotFile, altrimenti verrà triggerato l'evento GotText.
PutText(text text)	Esegue una richiesta http PUT utilizzando la proprietà url e la stringa dati passata come parametro. I caratteri della stringa dati saranno codificati mediante codifica UTF-8. Se la proprietà SaveResponse è settata a true, la risposta verrà salvata nel file indicato e verrà triggerato l'evento GotFile, altrimenti verrà triggerato l'evento GotText.
any JsonTextDecode(text jsonText)	Decodifica una stringa JSON. Molto utile per la decodifica delle risposte server.
any XMLTextDecode(text XmlText)	Decodifica una stringa XML. Molto utile per la decodifica delle risposte server.

Tabella 5 - Eventi, metodi e proprietà del componente Web.

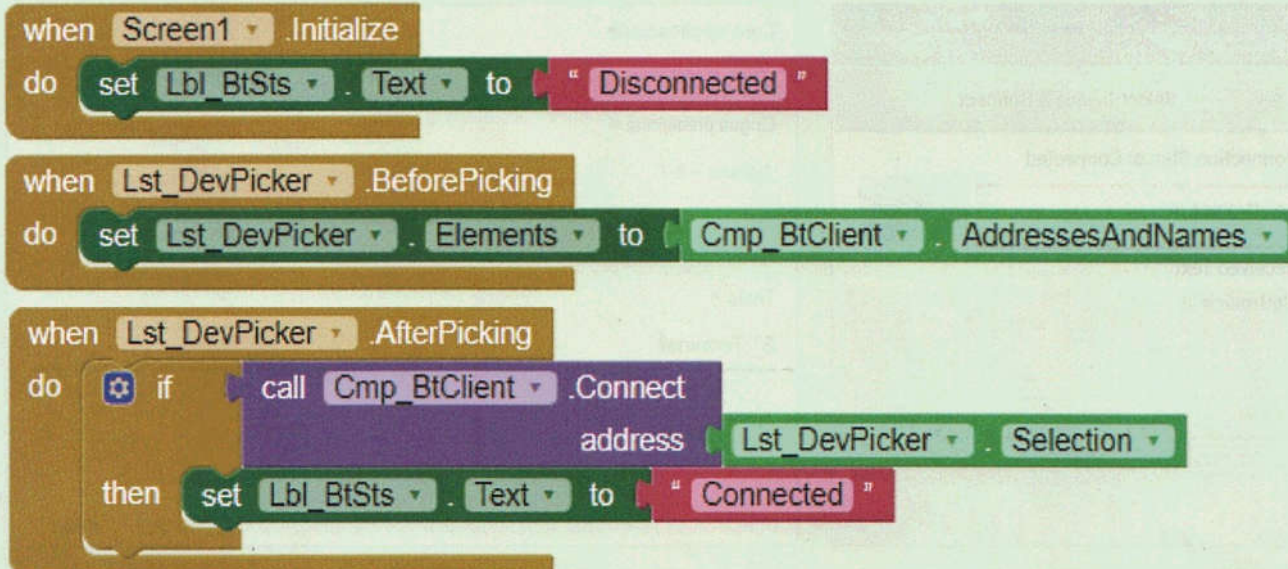


Fig. 5 - Inizializzazione e connessione con il dispositivo.

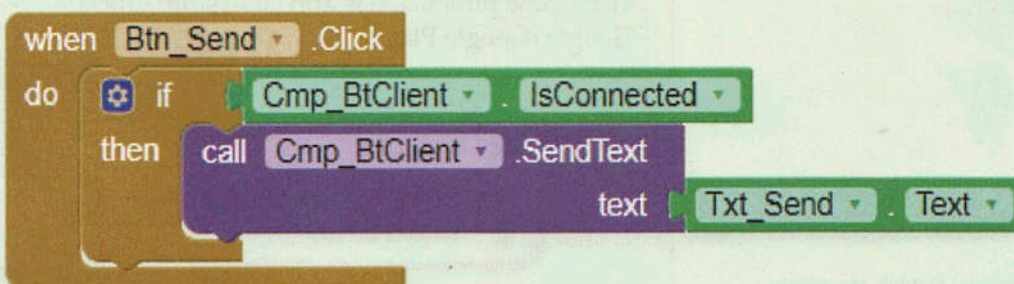


Fig. 6 - Invio dei dati.

cosa che può essere fatta dal seguente link: <https://play.google.com/apps/publish/signup/>.
 Notate che per consentirvi di effettuare la registrazione Google richiede una fee (un diritto, vale a dire la **Quota di registrazione**) di 25 dollari, quindi è necessario disporre di una carta di credito per effettuare il pagamento mediante il form proposto. La fee è a vita, quindi una volta che l'avrete corrisposta, non dovrete affrontare alcun costo aggiuntivo per ottenere un account da sviluppatore. Una volta ottenuto l'account da sviluppatore potete accedere alla Google Play Console ed iniziare la procedura per la pubblicazione della vostra app. Per iniziare la procedura cliccate sul pulsante "Pubblica un'applicazione Android su Google Play",

come illustrato in Fig. 9.

A questo punto si aprirà la finestra di pop-up di creazione dell'applicazione (Fig. 10) dove dovrete inserire la lingua predefinita ed il titolo, per poi cliccare sul pulsante "crea".

Dopo aver creato l'istanza per la distribuzione, bisogna compilare una per una le sezioni evidenziate in Fig. 11, ossia:

- scheda dello store;
- versioni dell'app;
- classificazione dei contenuti;
- prezzi e distribuzione.

Una volta completata una sezione comparirà l'icona con la spunta di colore verde a fianco. Tenete in considerazione che tutti i campi che è

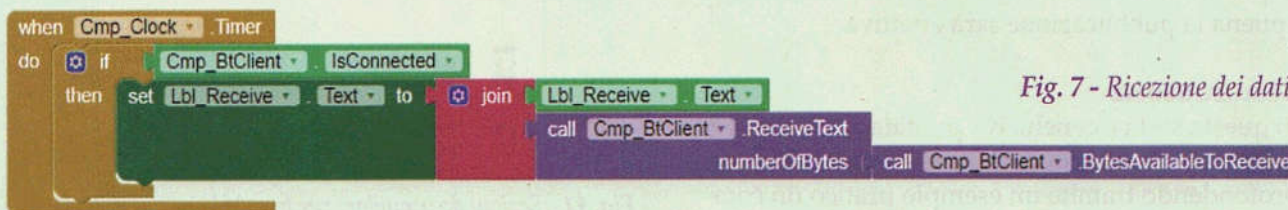


Fig. 7 - Ricezione dei dati.

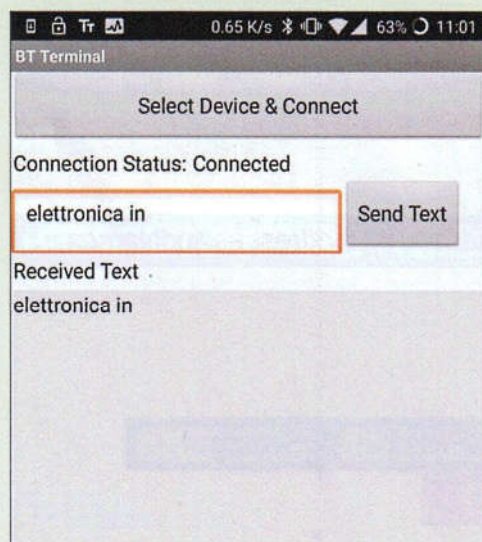


Fig. 8 - Applicazione di esempio.

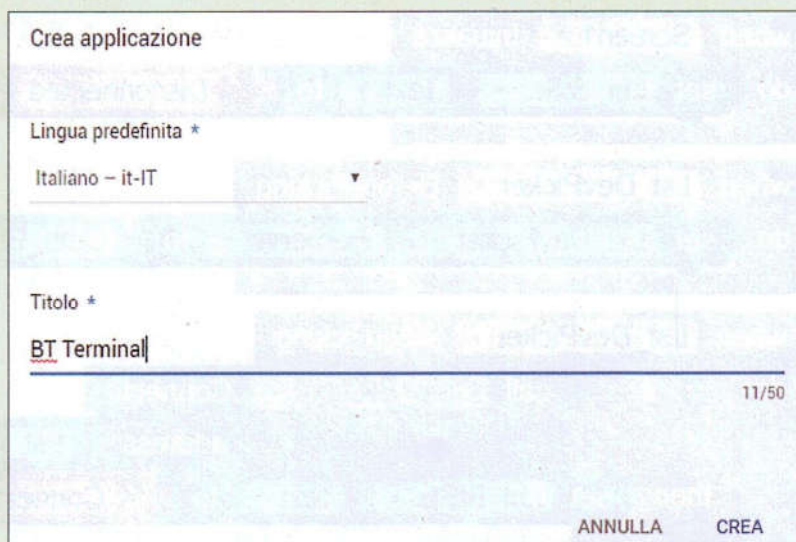


Fig. 10 - Finestra di pop-up di creazione applicazione.



Fig. 9 - Pulsante per avviare la procedura di pubblicazione.

obbligatorio compilare sono contrassegnati con un asterisco.

In Fig. 12 è rappresentato un esempio di compilazione dei campi iniziali della sezione "Scheda dello store".

Una volta riempite le varie sezioni bisogna tornare sulla sezione Versioni dell'App e cliccare sul pulsante blu "Esamina" in basso.

Nella schermata successiva troveremo un breve riepilogo e potremo pubblicare l'app cliccando sul pulsante "Inizia implementazione in versione di produzione", come illustrato in Fig. 13.

A questo punto l'app passerà nello stato "in attesa di pubblicazione" e Google Play vi notificherà non appena la pubblicazione sarà effettiva.

Conclusioni

In questa sesta e conclusiva puntata abbiamo visto le famiglie di componenti maps e connectivity, approfondendo tramite un esempio pratico un com-

ponente di quest'ultima famiglia. Inoltre abbiamo visto come pubblicare le app sullo store ufficiale di Google (Google Play).

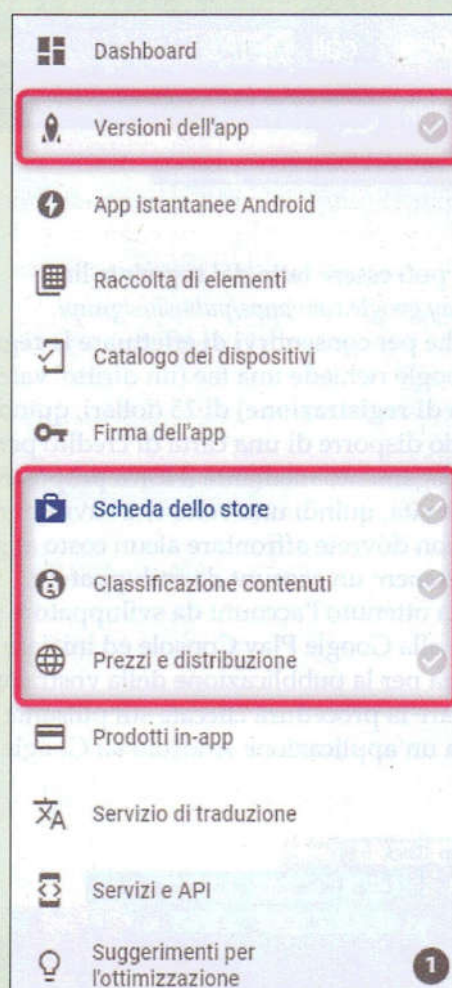


Fig. 11 - Sezioni da compilare per la pubblicazione sullo store.

Dettagli prodotto ITALIANO - IT-IT Gestisci traduzioni ▼

I campi contrassegnati con * devono essere compilati prima della pubblicazione.

Titolo *
Italiano - it-IT BT Terminal 11/50

Descrizione breve *
Italiano - it-IT Simple BT terminal 18/80

Descrizione completa *
Italiano - it-IT This application allows to communicate with a BT server device through SPP profile. 83/4000

Ti invitiamo a leggere le norme relative ai metadati per evitare alcune delle violazioni più comuni relative ai metadati delle app. Assicurati inoltre di rileggere tutte le altre norme del programma prima di inviare le tue app. Se la tua app o la tua scheda dello store sono idonee per i preavvisi al team responsabile dell'esame delle app di Google Play, contattaci prima della pubblicazione.

SALVA BOZZA

Fig. 12 - Campi iniziali della sezione scheda dello store.

Questa puntata completa la nostra panoramica su questo semplice ed efficace strumento di sviluppo per applicazioni Android e getta le basi per l'uti-

lizzo di App Inventor quale complemento di vari dispositivi elettronici gestiti da app e progetti futuri che vi proporremo.

APK in questa versione Espandi tutto

Tipo	Codice versione	Caricato	Dimensioni APK installato	Installazioni su dispositivi attivi
1 APK aggiunto				
▼ APK	1	9 minuti fa	2,94 MB	Nessun dato

Novità di questa versione

Ti consigliamo di aggiungere note per ogni nuova versione. In questo modo gli utenti potranno capire i vantaggi dell'upgrade all'ultima versione dell'app.

Torna indietro per aggiungere note sulla versione

INDIETRO **ELIMINA** **INIZIA IMPLEMENTAZIONE IN VERSIONE DI PRODUZIONE**

Fig. 13 - Pubblicazione dell'app sullo store.

Where is it?



trackimo

Novità



MINI LOCALIZZATORE SATELLITARE GPS/GSM CON FUNZIONE SOS, CONTROLLABILE DA SMARTPHONE E PC

Con Trackimo MINI, il più piccolo localizzatore della famiglia TRACKIMO, potrai tenere sotto controllo la posizione di persone, animali o cose, attraverso smartphone, tablet (iOS e Android) o PC. Trackimo racchiude un sistema di geolocalizzazione GPS/GSM e dispone di Bluetooth, Wi-Fi, antifurto e un pulsante SOS che potrai premere in caso di necessità, il tutto alimentato da una batteria al litio ricaricabile via USB. Viene fornito con una SIM da attivare gratuitamente con copertura MONDIALE per 12 mesi senza costi extra con allerte illimitate tramite APP ed e-mail. Trackimo MINI è anche impermeabile (IP65).

€ 149,00
Cod. **TRACKIMOMINI**

Prezzi IVA inclusa.

**CON TUTTI I PRODOTTI TRACKIMO HAI LA
SIM CON 12 MESI DI SERVIZIO INCLUSI!**

Rinnovo annuo € 49,00 oppure mensile € 4,90.



LOCALIZZATORE SATELLITARE GPS/GSM CONTROLLABILE DA SMARTPHONE E PC

**LUNGA
AUTONOMIA!**

€ 119,00
Cod. **TRACKIMO**



**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

La rete telefonica mobile di quinta generazione, ossia il 5G, debutterà nel 2022 e il suo arrivo, sebbene un po' tardivo, ridisegnerà i link radio per l'IoT. Questo perché consentirà connessioni ad altissima velocità: 100 Mbps in download e 50 in upload, con una latenza di 4 ms contro i 20 dell'attuale LTE.

L'impatto del 5G riguarderà energia e utility, manifattura, pubblica sicurezza, sanità, trasporto pubblico, media ed entertainment, automotive, finanza, retail e agricoltura.

L'avvento del 5G sarà importante per lo sviluppo dell'IoT in quanto consentirà la connessione di un numero di device molto più elevato di quanto non sia possibile oggi, con prestazioni molto più performanti che consentiranno un utilizzo più semplice e sicuro, anche in mobilità, di droni e robot. Infatti il 5G, essendo una rete cellulare,

Se il 5 potrà cambiare l'IoT



sfrutterà l'infrastruttura della telefonia mobile garantendo coperture più ampie e integrandosi con la telefonia stessa, come oggi fa l'LTE, ma garantendo throughput di dati più consistenti.

Il 5G potrà da una parte affiancare quelle già esistenti (WiFi, WiMax, SigFox, LoRa) e dall'altra estendere l'utilizzo dell'IoT. Al di là delle prestazioni, la connessione degli oggetti alla rete 5G consentirà di tracciare costantemente e identificare in tempo reale tutti gli oggetti collegati alla rete, semplificando le

problematiche legate agli standard per la comunicazione tra oggetti che utilizzano protocolli differenti. Per il collegamento dell'innumerabile quantità di oggetti prevista, sono stati avviati vari progetti, tra cui spicca quello di ARM, che ha realizzato una piattaforma software che consente equipaggiare con SIM virtuali i device IoT.

➤ VIA: il gateway per Alibaba Cloud IoT

Lo storico produttore di chipset VIA Technologies ha annunciato al mercato il progetto in grado di accelerare lo sviluppo di soluzioni Edge AI sicure e compatibili con l'ecosistema Alibaba Cloud IoT, come VIA ARTiGO A820, perfettamente integrabile nel progetto open source Link Edge di Alibaba Cloud IoT. Si tratta di un sistema fanless compatto dotato di connessioni dual Ethernet, una porta HDMI, una porta seriale COM RS-232/RS485 e predisposto per moduli opzionali wireless 3G e WiFi. Basato su SoC NXP i.MX 6DualLite ARM Cortex-A9 da 1.0GHz, il gateway per applicazioni IoT e industriali VIA ARTiGO A820, è stato progettato per supportare le installazioni IoT più esigenti grazie al design compatto e al risparmio energetico che ren-



de possibile. La soluzione VIA consente l'esposizione in Rete di una vasta gamma di sensori e collettori di dati eterogenei, grazie alle numerose interfacce e ai protocolli di comunicazione supportati.

<https://www.viatech.com/en/>

► Quando arriva il tram? Lo so con l'IoT

Cisco Systems, Intel e Dava Networks, in collaborazione con l'MTS (Metropolitan Transit System) di San Diego (California - U.S.A.) hanno realizzato un sistema di monitoraggio dei mezzi pubblici che tramite rete telefonica cellulare permette di fornire informazioni sulla posizione di tram e autobus con cui aggiornare quasi in real time i display informativi delle fermate del sistema di trasporti metropolitano della città.

Il progetto si basa sul sistema di Edge Analytics e rappresenta una soluzione di Edge intelligence in

grado di informare puntualmente i viaggiatori sui tempi d'attesa dei mezzi mediante i sistemi di segnaletica digitale installati alle varie fermate ed anche tramite specifiche app per tablet e smartphone. Inoltre gli stessi dati, uniti a quelli sulle condizioni operative dei mezzi, di diagnostica e manutenzione, vengono inoltrati allo staff tecnico dell'azienda di trasporti californiana, per garantire una maggiore sicurezza e funzionalità della rete tramviaria urbana attraverso, ad esempio, la sostituzione preventiva di parti di ricambio che vengono ritenute

► Google: il machine learning per tutti

Intelligenza artificiale e machine learning, sebbene vengano talvolta sovrapposti, sono due concetti distinti: intelligenza artificiale significa far sì che un computer imiti il cervello, mentre machine learning è un sottoinsieme dell'intelligenza artificiale e consiste nella capacità, per i computer, di apprendere. Entrambe le tecnologie sono fortemente incentivate da Google, che in occasione di Google Cloud Next 2018 ha annunciato di voler mettere data analytics e machine learning a disposizione di tutte le aziende, apportando una serie di aggiornamenti ai

servizi di data analytics e Cloud AI.

Per diffondere il machine learning Google conta su BigQuery ML, uno strumento che aiuterà il trasferimento dei dati da un data warehouse aziendale. BigQuery ML pone la potenza dei predictive analytics alla portata di milioni di utenti, anche quelli senza un background scientifico, portando il machine learning dove i clienti già memorizzano i propri dati.

BigQuery ML può essere considerato un'estensione di SQL a modelli scalabili su dataset di grandi dimensioni. Quindi por-

ta l'intelligenza artificiale dove vengono conservati i dati, semplificando il flusso di lavoro richiesto per creare funzioni evolute di machine learning.

Google supporta molti standard open per data analytics e machine learning. L'anno scorso l'azienda ha annunciato KubeFlow, per rendere più semplice l'uso di stack software di machine learning come TensorFlow, Scikit-Learn e altri. KubeFlow v0.2 è ora disponibile. Presenta un'interfaccia utente migliorata per navigare tra i componenti e numerosi miglioramenti al monitoraggio e alla creazione di report.

Il colosso di Mountain View offre anche Google Cloud, per il quale sono stati implementati strumenti come Cloud Deep Learning VM Image, che permette di creare progetti di machine learning utilizzando TensorFlow, scikit-learn e PyTorch.

<https://cloud.google.com>



Google BigQuery

prossime al guasto da sistemi basati su modelli matematici predittivi.

Il sistema, concepito nell'ambito del progetto Smart City Cisco e basato su router Cisco 829 e piattaforme e soluzioni IoT della californiana Davra Networks, sarà molto utile anche a garantire la sicurezza dei viaggiatori, in una rete estesa per circa 100 km, che ogni anno garantisce 92 milioni di corse, quindi difficile da monitorare con la presenza materiale di personale di vigilanza e tecnici. Un esempio che certamente ispirerà altri sistemi di trasporto pubblico nel mondo.

www.cisco.com



» Da Samsung, il nuovo SoC per la IoT

SAMSUNG
Exynos i



Samsung è arrivata alla sua seconda generazione di chip dedicati all'Internet of Things: si tratta del nuovissimo SoC Samsung Exynos i S111, la cui sezione radio (funzionante a banda stretta) riesce ad offrire una copertura ancora più ampia rispetto al passato, consumi ulteriormente ridotti. Il SoC integra modem, microprocessore, memoria ed un ricevitore per il posizionamento satellitare compatibile Global Navigation Satellite System (GNSS) quindi multistandard. Utilizzando il supporto alla rete

LTE Rel.14, il data-rate può raggiungere i 127 kbps/s in download e 158 kbps in upload. Inoltre, grazie a una nuova modalità di risparmio energetico, il chip può operare per lunghi periodi senza interruzioni, comunicando su lunghe distanze, all'interno di grandi edifici. Ciò viene ottenuto grazie a un meccanismo di ritrasmissione dei dati continuo, fino al trasferimento completo o fino ad un numero prestabilito di ritrasmissioni in caso di errore. Eseguendo una sessione di ritrasmissione continua, il SoC Exynos i S111 può coprire una distanza pari a 10 chilometri (km) o più, risultando quindi adatto a sistemi IoT remoti per grandi aree.

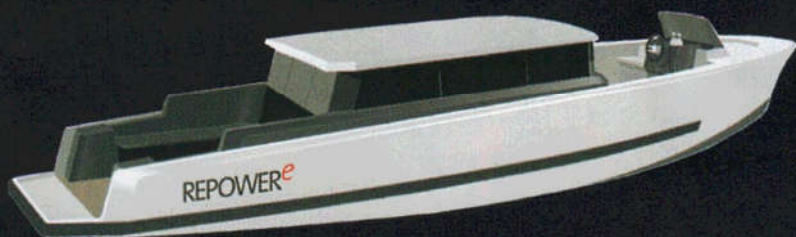
Il ricevitore Global Navigation Satellite System (GNSS) è capace di determinare la posizione quasi in tempo reale, con un'elevatissima precisio-

ne, grazie alla combinazione dei dati di più sistemi. La precisione viene aumentata da un sistema complementare di posizionamento chiamato OTDOA (Observed Time Difference of Arrival), una tecnica di rilevamento della posizione complementare a quello che utilizza i ripetitori di rete mobile con cui è possibile, tra l'altro, sopperire alla momentanea carenza del segnale satellitare. Il SoC Samsung implementa anche accorgimenti per la sicurezza dei dati trasmessi ottenuta da un hardware chiamato Security Sub System ed il PUF (Physically Unclonable Functions) IP, che protegge i dati garantendo l'autenticità degli stessi.

Samsung Exynos i S111 sarà disponibile entro la fine dell'anno.

www.samsung.com

Repower presenta la sua prima imbarcazione "full electric" al Salone Nautico di Genova



Repower ha presentato alla 58^a edizione del Salone Nautico di Genova la sua ultima creazione, *Repower*, una delle prime barche nate "full electric". *Repower* è l'evoluzione cabinata di una imbarcazione aperta, ricaricabile anche attraverso sistemi di carica veloce del mondo e-cars come i sistemi Repower PALINA e BITTA. Con *Repower* viene sviluppato coerentemente il proprio approccio alla mobilità elettrica, che prende il nome di Homo Mobilis by Repower e si compone di diversi prodotti, soluzioni digitali e modelli innovativi che sono diventati best practice.

Di strada nel mondo della mobilità sosteni-

nibile ne è stata fatta molta dopo la consegna di una delle prime auto elettriche in Italia nel 2010, taggata "powered by Repower". Oggi la multinazionale dell'energia rappresenta uno dei player di riferimento del settore, potendo contare su un ventaglio di soluzioni innovative, sotto il profilo dell'approccio, dell'innovazione e del design, tra cui spiccano gli strumenti di ricarica PALINA e BITTA. Oggi la nautica rappresenta una nuova frontiera in questo campo e Repower è pronta a dare il suo contributo, presentando la sua visione di sostenibilità a 360°.

www.repower.com

Spedizione Polarquest, alla ricerca di microplastiche in Artico

In 18 giorni di navigazione la barca a vela Nanuq impegnata nella spedizione Polarquest2018 ha percorso 1500 miglia: si è conclusa così la spedizione Polarquest2018, il progetto di cui il Cnr è partner scientifico, promosso in occasione del novantesimo anniversario della storica spedizione al polo di Umberto Nobile. La barca a vela Nanuq, impegnata nella spedizione, ha completato con successo lo scorso 22 agosto la circumnavigazione dell'arcipelago delle Svalbard, raggiungendo le acque al largo di Longyearbyen (Isfjord), dove la tappa aveva preso inizio lo scorso 4 agosto. Tra gli scopi scientifici della spedizione, anche quello di raccogliere campioni di acque alle latitudini più estreme, con l'obiettivo di verificare la presenza di microplastiche e microfibre: ad effettuare le analisi sui campioni saranno i laboratori di Lerici (La Spezia) dell'Istituto di scienze marine (Ismar) del Cnr. "Diversi modelli indicano che nella zona artica po-

La tappa italiana dell'Energy Observer, il primo catamarano a idrogeno al mondo

Impegnato in un giro del mondo della durata di 6 anni, questa estate l'Energy Observer ha fatto tappa nel Mediterraneo toccando anche alcuni porti italiani tra i quali Venezia. Primo catamarano a idrogeno, autosufficiente e a zero emissioni, l'imbarcazione è impegnata nella circumnavigazione del globo con 101 tappe in 50 paesi diversi. Scopo dell'impresa

dimostrare come sia possibile effettuare viaggi anche molto lunghi utilizzando esclusivamente energie rinnovabili, nel caso specifico l'idrogeno, ottenuto per elettrolisi grazie all'energia fornita da 141 mq di pannelli fotovoltaici.

L'idrogeno viene stoccato in appositi serbatoi ed utilizzato per alimentare alcune celle a combustibile che generano l'e-





trebbe accumularsi una quantità notevole di detrito marino: ovviamente la sorgente non è locale, ma proviene da lontano, potrebbe trattarsi di materiale originato nell'oceano Atlantico, così come di residui trasportati dai fiumi siberiani o addirittura dallo Stretto di Bering. Un dato è certo: neppure l'Artico, ormai, è esente da questo problema", afferma il responsabile Cnr-Ismar di La Spezia Stefano Aliani.

A bordo del Nanuq anche un rivelatore di raggi cosmici assemblato al Cern: i suoi dati permetteranno di studiare l'influenza dei raggi cosmici nei processi di formazione delle nubi, e di migliorare la comprensione dei cambiamenti climatici mettendo in correlazione le Supernovae con le fasi climatiche su un periodo di oltre 500 milioni di anni.

www.cnr.it

lettricità necessaria alla propulsione.

Il grande scafo di 30,5 per 12,8 metri, ricavato dal refitting di una barca a vela degli anni '70, è rimasto ormeggiato all'Isola della Certosa, "paradiso" naturale della Laguna per il diporto e l'ambiente. Qui il team degli 11 membri dell'equipaggio e gli sponsor tecnici hanno allestito un Villaggio con un percorso didattico.

Dalla partenza a Saint-Malo (Francia) nel giugno 2017, Energy Observer ha percorso più di 7.600 miglia lungo le principali rotte commerciali marine, e terminerà la sua "Odissea per il futuro" nel 2022, con l'arrivo a New York alle Nazioni Unite.

www.energy-observer.org



Inaugurato in Emilia il primo impianto biogas bi-stadio europeo



È stato realizzato in Italia il primo impianto a biogas bi-stadio europeo che sfrutta la digestione anaerobica per produrre energia.

Lo ha costruito Biogas Italia a Soliera (Modena) grazie all'impiego di una tecnologia brevettata da ENEA e CREA, che consente di produrre il 20% in più di energia rispetto alle strutture "tradizionali". "Nell'impianto bi-stadio, le prime fasi del processo (idrolisi e acidogenesi) vengono separate dalle restanti (acetogenesi e metanogenesi) in reattori distinti, dove agiscono ceppi batterici che operano in modo ottimale con tempi e valori di pH diversi", spiega Vito Pignatelli, responsabile del laboratorio Biomasse e Biotecnologie per l'Energia dell'ENEA. Grazie a tempi più rapidi di digestione della biomassa, la tecnologia bi-stadio consente di realizzare impianti a biogas più piccoli ed economici rispetto

a quelli tradizionali, in quanto impiega fino al 15% in meno di materia prima, a parità di energia prodotta.

"Dimensioni, efficienza e costi ridotti sono i tre elementi con i quali questa tecnologia potrà contribuire all'ulteriore diffusione in Italia del biogas, una risorsa fondamentale per il processo di decarbonizzazione del sistema energetico nazionale", conclude Pignatelli. Oltre al contenuto tecnologico, l'impianto di Soliera testimonia la virtuosa collaborazione tra ricerca pubblica e impresa con ENEA e CREA concessionarie dell'uso del brevetto alla Biogas Italia srl, che a sua volta ha beneficiato di un finanziamento di Invitalia Spa e si è accordata con l'azienda agricola Lugli per l'utilizzo del terreno e lo sfruttamento delle materie prime per alimentare l'impianto.

www.enea.it



Impianto fotovoltaico da 26 MWp alle Saline Conti Vecchi

Eni e GSE hanno presentato alle Saline Conti Vecchi, l'impianto fotovoltaico della capacità di 26 MWp in fase di realizzazione presso il polo industriale di Assemini in provincia di Cagliari. L'impianto rientra nell'ambito del Progetto Italia, l'insieme di iniziative che Eni sta realizzando allo scopo di valorizzare, in ottica sostenibile, le proprie aree industriali dismesse, in particolare nel Mezzogiorno. L'evento rappresenta il frutto dell'accordo di collaborazione Eni-GSE, siglato lo scorso 6 giugno 2017, con il quale le due società hanno dato seguito agli impegni presi implementando tecnologie per la produzione di energia elettrica da fonti rinnovabili, va-

lorizzando il territorio attraverso il rilancio dell'occupazione e supportando l'economia sostenibile. Grazie ad una potenza massima di 26 MWp, l'energia prodotta dai pannelli solari installati andrà ad alimentare il ciclo di produzione del sale, prodotto dalla società Ing. Conti Vecchi, coprendo circa il 70% del consumo energetico totale. In questo senso, Eni e GSE sono in prima fila nel supportare politiche di economia circolare che promuovano uno sviluppo sostenibile attraverso la riduzione delle emissioni di CO2 e il riutilizzo del suolo.

www.eni.com



Inaugurata la "Casa Solare" del Politecnico di Torino in gara in Cina

In tre settimane hanno costruito una casa vera, pronta per essere abitata e per di più con caratteristiche di autonomia energetica e sostenibilità uniche: sono i 17 studenti di ingegneria e architettura del Politecnico di Torino che hanno lavorato insieme ai loro colleghi della South China University of Technology (SCUT) di Guangzhou per costruire la "casa solare" che competerà alla Solar Decathlon China 2018. A questo contest internazionale di Architettura competono 21 progetti delle principali scuole di architettura mondiali. I numeri dei visitatori attesi per l'esposizione dei progetti - che si trasformerà, al termine dell'evento, in un quartiere modello - sono notevoli: circa 2 milioni di visitatori in loco e 100 milioni di contatti web previsti. Le giornate di visita libera si alterneranno alle prove di efficienza cui gli edifici saranno sotto-

posti, le quali determineranno nel loro complesso - insieme ad un giudizio sulla qualità architettonica delle realizzazioni - la graduatoria finale. Si tratta di uno dei più importanti concorsi internazionali di architettura, riservato a team congiunti di docenti e studenti delle principali scuole di architettura a livello globale.

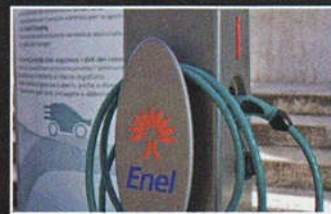


Auto elettrica: in arrivo da Enel altre 14.000 colonnine

La Banca europea per gli investimenti (BEI) è al fianco di Enel nel piano di installazione di circa 14.000 colonnine di ricarica per i veicoli elettrici in tutta Italia nei prossimi cinque anni. È questo l'obiettivo dell'accordo siglato a Roma tra la BEI ed Enel X Mobility, controllata di Enel X, l'Unità del Gruppo Enel dedicata allo sviluppo di prodotti innovativi e soluzioni digitali. Enel X Mobility sarà infatti destinataria di un finanziamento complessivo di 115 milioni di euro e della durata di 10 anni, garantito da Enel Spa, di cui 50 milioni di euro già perfezionati. Enel X è impegnata a dare un forte impulso allo sviluppo della mobilità elettrica in Italia con il suo Programma nazionale che punta a dotare il Paese di 7mila punti di ricarica entro il 2020, fino ad arrivare a 14.000 entro la fine del 2022, investendo circa 300 milioni di euro.

La rete capillare di ricarica di Enel X sarà composta da colonnine Quick (22 kW) nelle aree urbane e Fast (50 kW) e Ultra Fast (oltre 150 kW), per la ricarica veloce, in quelle extraurbane.

www.enel.com



Il contest prevede la progettazione e la costruzione di una residenza monofamiliare innovativa ad alte prestazioni energetiche, alimentata ad energia solare, che verrà quindi sottoposta ad una serie di test di performance, che contribuiranno alla composizione della graduatoria finale.

www.polito.it





€ 89,00

Cod. **ROBOARM**

**Entra
nell'affascinante
mondo della
robotica!**

**PICK AND PLACE PER BRACCIO
ROBOTICO "ROBOARM"**

BRACCIO ROBOTICO IN PLEXIGLASS - IN KIT

Costruire un braccio robotico è un'esperienza interessante per chi vuole avvicinarsi al mondo della robotica e prepararsi a sfide più complesse. Questo kit contiene tutto il necessario per realizzarne uno di tipo articolato in plexiglass a 4 gradi di libertà, con base rotante (180°), pinza per afferrare gli oggetti (montabile sia in posizione orizzontale che verticale) e 5 servocomandi gestibili da Arduino o altra elettronica (acquistabile separatamente). È disponibile anche il kit "Pick and Place" composto da una piccola pompa a depressione, un'elettrovalvola, un microswitch, cavi, tubi e ventosa, che permette di equipaggiare il braccio robotico con un sistema di prelievo e rilascio di piccoli oggetti.



€ 45,00

Cod. **PEPARM**

Scopri anche gli altri modelli ...

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



**BRACCIO ROBOTICO
CON CONTROLLO
VIA CAVO - IN KIT**

€ 54,00

Cod. **KSR10**

**BRACCIO ROBOTICO
6DOF CON PINZA E
SERVI RC - IN KIT**



€ 129,00

Cod. **6DOFARM**

Completo di pinza per afferrare gli oggetti.
Dotato di servi RC con ingranaggi in metallo.

**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

PROMOSSO E ORGANIZZATO DA



CON IL PATROCINIO DI



ROMA



CON IL SUPPORTO DI



IN COLLABORAZIONE CON



REGIONE
LAZIO

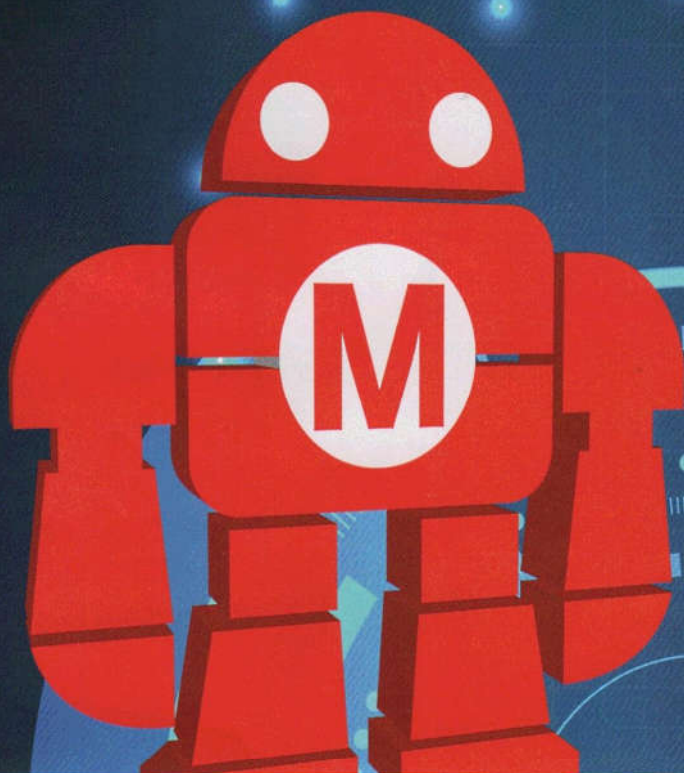
ROME

Maker Faire®

THE EUROPEAN EDITION

Hands on Innovation!

Business, Networking, Education, Entertainment



OCTOBER
12 - 14

FIERA DI ROMA

MAKERFAIREROME.EU

Instagram Facebook Twitter #MFR18

MAIN PARTNER



GOLD PARTNER

ARROW

FERROVIE
DELLO STATO
ITALIANE

aceq

UNIDATA

MEDIA PARTNER

Focus

Focus Junior

TECHNICAL
PARTNER

FUTURA
ELETTRONICA
futurashop.it

OFFICIAL RADIO

RDS

OFFICIAL CARRIER

TRENITALIA

LOCATION

FIERA
DI ROMA

THANKS TO

Make: