

Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXV - n. 236 - GIUGNO 2019

Driver per motori passo-passo con encoder

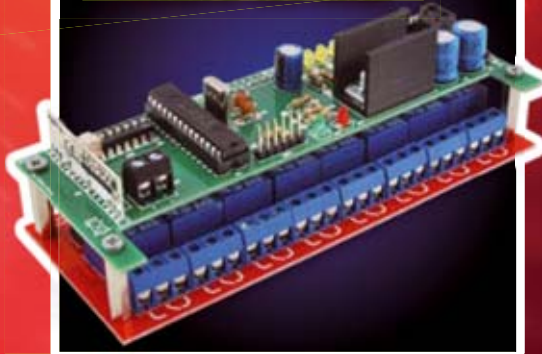


- Speciale celle di carico
- Modem cellulare USB
- Centralina di irrigazione Bluetooth
- Controlliamo la TV con Fishino e Google Home
- CCTV con Raspberry Pi e MotionEye

Amplificatore stereo universale



Radiocomando 8 canali a 433 MHz



UNI-T® STRUMENTI DI MISURA PER OGNI ESIGENZA!



TACHIMETRO DIGITALE CON LASER E INTERFACCIA USB PER PC

€78,00

Cod. TP0030

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



MINI FONOMETRO DIGITALE

€ 25,00

Cod. TP0070

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

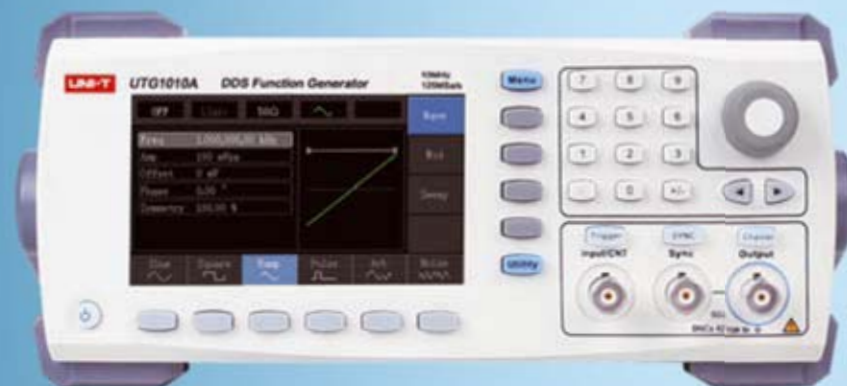


MULTIMETRO DIGITALE CON MISURA DI TEMPERATURA, CAPACITÀ, FREQUENZA, E TRANSISTOR

€31,90

Cod. TP0060

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



€ 278,00

Cod. TP0073

GENERATORE DI FUNZIONI/FORME D'ONDA ARBITRARIE - 10 MHZ

Generatore di funzioni e forme d'onda arbitrarie a 1 canale, con display TFT LCD da 4,3 pollici, frequenza di campionamento 125 MS/s, frequenza max. di uscita 10 MHz. Dispone di connessione USB. Grazie alla tecnologia DDS, è in grado di generare forme d'onda precise e stabili con una risoluzione fino a 1 μHz. Forme d'onda standard: sinusoidale, quadra, rampa, triangolare, impulso, noise, DC, arbitraria. Supporta le modulazioni AM, FM, PM, PWM, ASK, FSK, PSK.

Campi di applicazione: sensori analogici, segnali di ambiente di simulazione, test di circuiti, test IC, ricerca e formazione.



TERMOMETRO DIGITALE PORTATILE PER TERMOCOPPIE TIPO K E J

€ 19,90

Cod. TP0062

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



LUXMETRO PORTATILE 0 - 199900 LUX

€ 34,00

Cod. TP0063

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



ANEMOMETRO DIGITALE

€ 37,00

Cod. TP0064

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

Prezzi IVA inclusa



AI e deep learning, un primo importante successo

Famoso per le sue traduzioni poco accurate e talvolta imbarazzanti, tanto da essersi attirato nel corso degli anni – proprio a causa degli errori di traduzione più bizzarri – ogni genere di commenti, Google Translate si sta riprendendo la rivincita con traduzioni sempre più valide, paragonabili a quelle dei più bravi traduttori in carne ed ossa. Basta provare, per rendersi conto di ciò.

Tutto è iniziato poco più di due anni fa quando Google ha annunciato una modifica radicale del programma utilizzato, ovvero il passaggio dal precedente sistema "phrase-based" al nuovo Google Machine Neural Translation (GMNT).

La precedente versione di Google Translate considerava una parola per volta, per poi tradurla e riordinare i lemmi nell'ordine grammaticale corretto con le regole sintattiche e semantiche proprie

di ogni lingua che venivano ricavate dal sistema grazie all'analisi di un altissimo numero di testi già tradotti e memorizzati.

L'enorme numero di eccezioni tipiche di ogni lingua e l'intrinseca ambiguità di

molte parole rendevano però poco preciso il risultato.

Il nuovo Google Translate, invece, non analizza le singole parole, ma guarda al contesto: selezionando intere sequenze di frasi come unità di base, impara a tradurre grazie ad un iniziale training in cui propone traduzioni e riceve dai programmatori dei feedback sulla precisione del risultato.

Grazie alle migliaia di traduzioni fornite, il programma individua il significato di ogni espressione più adatto a ciascun contesto.

È proprio questo tipo di apprendimento basato sull'esperienza a rendere l'intelligenza artificiale simile al cervello umano, che funziona più su base intuitiva-deduttiva che non utilizzando regole pre-programmate.

In questi due anni Google Translate ha migliorato continuamente le proprie capacità, facendo tesoro dell'esperienza quotidiana e fornendo traduzioni sempre più performanti. Oggi, secondo un sistema oggettivo di valutazione delle traduzioni, c'è ancora un leggerissimo vantaggio dell'uomo sulla macchina, ma è solo questione di tempo.

Gli sviluppi di una efficiente traduzione automatizzata sono ancora tutti da scoprire, anche perché il passaggio al "parlato" è praticamente immediato, con tutti i risvolti del caso.

Per il momento prendiamo nota di questa prima, convincente, prova dell'intelligenza artificiale.

Il nuovo Google Translate non analizza le singole parole ma guarda al contesto.

ON LINE

elettronica.in

SI RINGRAZIA

Futura Elettronica
Microchip
Mouser
RM Elettronica



In copertina:
Controller a retroazione per motore passo-passo in grado di verificare dinamicamente il posizionamento dell'albero e prevenire eventuali perdite di passo.

Arsenio Spadoni
 arsenio.spadoni@elettronica.in



Elettronica In

Rivista mensile, anno XXV n. 236
GIUGNO 2019

Direttore Responsabile

Arsenio Spadoni

arsenio.spadoni@elettronica.in

Redazione

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

redazione@elettronica.in

Grafica

Alessia Sfulcini

alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

monica.premoli@elettronica.in

Ufficio Abbonamenti

Elisa Guarniero (0331-752668)

elisa.guarniero@elettronica.in

Direzione, Redazione, Pubblicità

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

Tel. 0331-752668

Abbonamenti

Annuo 10 numeri Euro 45,00

Esteri 10 numeri Euro 45,00 (digitale)

Le richieste di abbonamento vanno inviate a:

FUTURA GROUP srl

via Adige 11, 21013 Gallarate (VA)

Tel. 0331-752668

oppure tramite il sito

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

Distribuzione per l'Italia

SO.DI.P. Angelo Patuzzi S.p.A.

via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

ROTO3 Spa - Via Turbigo, 11/b

20022 CASTANO PRIMO (MI)

Tiratura
18.500 copie



Mensile
associato all'USPI,
Unione Stampa
Periodica Italiana

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

Prezzo di copertina Euro 6,00.

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

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

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

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

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

© 2019 Futura Group srl



Contenuti

01

RUBRICHE

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

Progetto di Copertina

MOTORFISH: LO STEPPER NON SBAGLIA UN PASSO

46



Fishino

Fishino IR
remote

23



23

ARTICOLI

AUTOMAZIONE

23

Fishino IR remote

Uniamo la potenza di Fishino UNO alla comodità degli assistenti vocali Google Home e potremo dimenticarci del telecomando e gestire vari dispositivi con delle semplici frasi.

WIRELESS

37

Radiocomando 8 CH a 433MHz

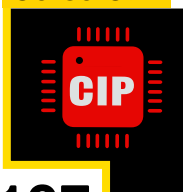
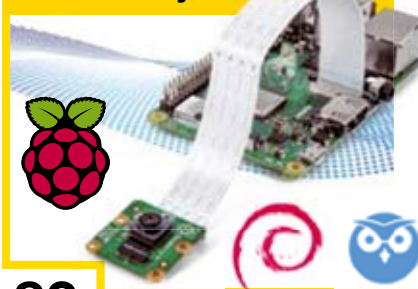
Ricevitore UHF ad autoapprendimento composto da una scheda radioricevente con decodifica Motorola e un modulo dotato di 8 relé operanti in modalità impulsiva o bistabile.

Wireless

Radiocomando
8 CH a 433MHz



37

Didattica**Modem
cellulare USB****67****Domotica****Irrigazione a
controllo BT****57****Audio****Amplificatore stereo
universale****75****Didattica****Celle di carico****97****Didattica****Corso CIP****107****Raspberry Pi****CCTV con Raspberry Pi
e Motion Eye****83****46****MONDO 3D****MOTORFISH:
lo stepper non
sbaglia un passo**

Controller a retroazione per motore passo-passo in grado di verificare dinamicamente il posizionamento dell'albero e prevenire eventuali perdite di passo.

DOMOTICA**57****Irrigazione a controllo BT**

Analizziamo il software del progetto e l'interfaccia utente, che è costituita da un'App sviluppata con Android Studio. Una mappa messaggi ben definita determina come app e firmware devono scambiarsi le informazioni. Seconda e ultima puntata.

DIDATTICA**67****Modem cellulare USB**

Realizziamo un dispositivo che consente la connettività GPRS tramite la porta USB di qualsiasi PC o dispositivo host USB.

AUDIO**75****Amplificatore stereo universale**

Composto da una scheda base su cui montare un modulo finale a scelta tra LM4835 o PAM8403, permette di realizzare amplificatori audio per ogni applicazione, sia per cuffie che per pilotare una coppia di altoparlanti.

83**RASPBERRY PI****CCTV con Raspberry Pi e MotionEye**

Costruiamo il nostro sistema centralizzato di videosorveglianza domestico controllato via web. Ultima puntata.

97**DIDATTICA****Celle di carico**

Conosciamo i sensori e l'elettronica necessaria per la misura del peso di oggetti in moltissimi ambiti, dalle bilance elettroniche alle pesche industriali.

107**DIDATTICA****Conoscere ed usare le Core
Independent Peripherals**

Concludiamo la nostra carrellata sulle CIP, studiando quella che si trova solo sui microcontrollori di fascia più alta, siano esse a 8, 16 o 32 bit, ovvero quella relativa al DMA. Settima e ultima puntata.



Differenzia la Tua Applicazione

Personalizza il Tuo Progetto Embedded con i Flessibili ed Integrati Building Block

Microchip comprende che le tue esigenze di progettazione sono uniche. Ecco perché i nostri prodotti offrono un robusto set di periferiche integrate per un'ampia varietà di applicazioni:

- Aggiungi rilevamento tattile e display accattivanti e di impatto offrendo una interazione intuitiva all'utente
- Connetti al resto del mondo la tua applicazione, wired o wireless
- Controllo motori o della conversione di potenza
- Proteggi i dati delle tue applicazioni

Personalizza il tuo prodotto con periferiche integrate e riduci costi e tempi di progettazione.

Personalizza il tuo progetto, vai su
www.microchip.com/FlexibleFunctions





Comportamento dei cavi al fuoco

Sto rifacendo l'impianto elettrico di una vecchia casa ma ho avuto qualche difficoltà nel scegliere i cavi da utilizzare. Vorrei in ogni caso utilizzare dei cavi autoestinguenti ...

Roberto Sassi > Roma

R: La definizione di un cavo autoestinguente è puramente convenzionale, dal momento che è determinata dai risultati di prove specifiche come previsto dalle norme. La proprietà di non propagazione alla fiamma di un cavo dipende da molti fattori: dal materiale utilizzato, dal suo spessore, dal suo peso per metro lineare e, soprattutto, dal sistema di installazione.

Senza voler affrontare fino a fondo l'argomento che richiederebbe molto più spazio, diciamo che in funzione del loro comportamento al fuoco, i cavi elettrici possono essere: non propaganti la fiamma; non propaganti l'incendio; a bassa emissione di fumi, gas tossici e corrosivi (LSOH); resistenti al fuoco. Vediamo le differenze.

Un cavo "*non propagante la fiamma*", è dotato di un isolamento autoestinguente, ma è efficace soltanto se è installato da solo; se i cavi non propaganti la fiamma vengono installati in fascio, l'incendio si propaga lungo il fascio stesso. Due cavi distanti meno di 25 cm costituiscono già un fascio. L'isolamento di un cavo "*non propagante l'incendio*", invece, è autoestinguente anche quando è posato in fascio.

È bene ricordare che se i cavi non propaganti l'incendio sono installati in numero maggiore di quello previsto dalle norme di prodotto, non è più garantita l'autoestinguenza del fascio. Un cavo LSOH (Low Smoke Zero Halogen) ha un isolamento che brucia con bassa emissione di fumi, gas tossici e corrosivi. Un cavo "*resistente al fuoco*" deve continuare a funzionare, per un tempo stabilito dalle norme, anche se sottoposto al fuoco.

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

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

Contattateci all'indirizzo: redazione@elettronica.it

Cavi non propaganti la fiamma

L'isolamento di un cavo non propagante la fiamma può bruciare, ma solo per una lunghezza non superiore a quella indicata nella specifica norma di prodotto; in assenza di indicazioni della norma si assume una lunghezza massima di 55 cm.

Nelle condizioni di prova stabilite dalle norme EN 60332 (CEI 20-35), il cavo viene sottoposto alla fiamma di un bruciatore a gas. Quando si spegne il bruciatore, il cavo deve smettere di bruciare e non deve lasciar cadere gocce di isolante incandescente. Sigle

di alcuni tipi di cavi non propaganti la fiamma: H05V-K, H07V-K, H05VV-F, H05RN-F, H07RN-F, H07RN8-F.

Cavi non propaganti l'incendio

Le condizioni di prova per un cavo non propagante l'incendio sono stabilite dalle norme CEI 20-22. Il fascio di cavi, di dimensioni indicate nelle norme di prodotto, viene installato in verticale e sottoposto alla fiamma di un bruciatore a gas; l'incendio non si deve propagare lungo il fascio per oltre 2,5 m. Sigle di alcuni tipi di cavi non propaganti l'incendio: N07V-K, FROR 300/500 V, FROR 450/750 V, FG7(O)R 0,6/1 kV.

Cavi LSOH (Low Smoke Zero Halogen)

I cavi LSOH vengono sottoposti alla "prova dei fumi" secondo le norme EN 50267 (CEI 20-37). Sono richiesti negli ambienti di pubblico spettacolo o nei casi in cui i fumi di un incendio possono causare danni irreparabili alle cose, ad esempio in un museo. Se i cavi sono installati sotto traccia o dentro tubi o canaline con grado di protezione IP4X, i cavi LSOH non sono necessari. Sigle di alcuni tipi di cavi LSOH non propaganti l'incendio: N07G9-K, H07Z1-K (tipo 2), FM9 450/750 V, FG7(O)M1 0,6/1 kV, FG10(O)M1 0,6/1 kV.

Cavi resistenti al fuoco

Le prove a cui deve essere sottoposto un cavo resistente al fuoco sono stabilite dalle norme CEI 20-35; il cavo viene sottoposto sia alle fiamme di un bruciatore, sia a shock meccanici. Un cavo resistente al fuoco è provvisto di una nastratura a base di vetro e mica sul conduttore, in conformità alla norma CEI 20-45, oppure utilizza un isolamento minerale (una guaina esterna in rame racchiude i conduttori isolati con ossido di manganese), in conformità alla EN 60702 (CEI 20-39). Sigla cavo LSOH non propaganti l'incendio e resistente al fuoco: FTG10(O) M1 0,6/1 kV. Per concludere ricordiamo che per garantire l'alimentazione durante un incendio, si può utilizzare un cavo ordinario, interrato o protetto da uno schermo REI 60.

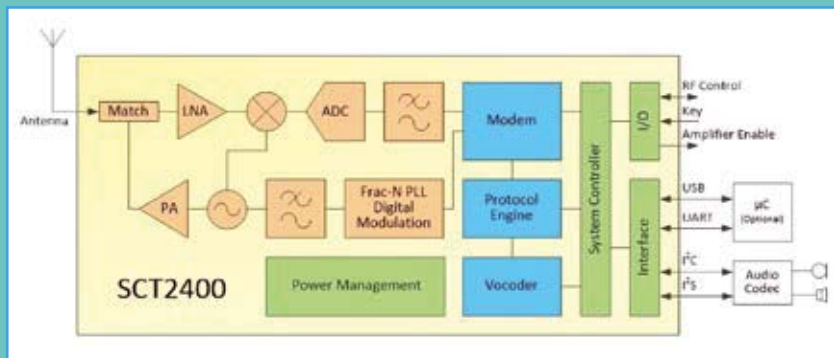


Un RTX audio e dati sui 2,4 GHz

Ho intenzione di realizzare un RTX audio codificato con dimensioni particolarmente ridotte. Cosa mi consigliate?

Luca Marnati > Milano

R: Esistono varie possibilità per realizzare quanto richiesto; una delle più innovative – non foss'altro perché appena introdotta sul mercato – è quella di CML che ha annunciato la disponibilità del ricetrasmittitore SCT2400, progettato per supportare canali di comunicazione voce e dati digitali sicuri a lungo raggio (da 2 a 12 km in portata ottica) per sistemi che utilizzano bande senza licenza. Il chip SCT2400 misura appena 10 x 10 mm e richiede pochissimi componenti esterni; si tratta di un ricetrasmittitore radio digitale altamente integrato operante nella gamma di frequenze di 2,4 GHz. La sua eccezionale portata di oltre 12 km (portata ottica) con una potenza di uscita fino a 100 mW si ottiene consumando solo 10 mA in ricezione, circa 52 mA in ricezione con audio attivo e 64 mA in trasmissione. Le sue caratteristiche di sicurezza e basso consumo ne fanno una soluzione ideale per una vasta gamma di applicazioni digitali per voce e dati. Le funzionalità integrate in SCT2400 includono la codifica vocale di CML con la correzione degli errori, il che significa che il dispositivo può mantenere la chiarezza e la sicurezza su lunghe distanze in sistemi radio a bassa potenza come Private Mobile Radio (PMR) e Land Mobile Radio (LMR). Le ridotte dimensioni lo rendono utilizzabile in prodotti



portatili, mobili e indossabili, inclusi tracker fitness, abbigliamento smart e cuffie wireless. SCT2400 opera da una singola alimentazione a 3 V e supporta le modalità vocali digitali che includono peer to peer e chiamata di gruppo.

Le Classi del protocollo LoRaWAN

Ho letto che il protocollo LoRaWAN prevede tre Classi per la comunicazione tra dispositivo remoto e gateway. In cosa si differenziano queste Classi?

Lorenzo Urbani > Firenze

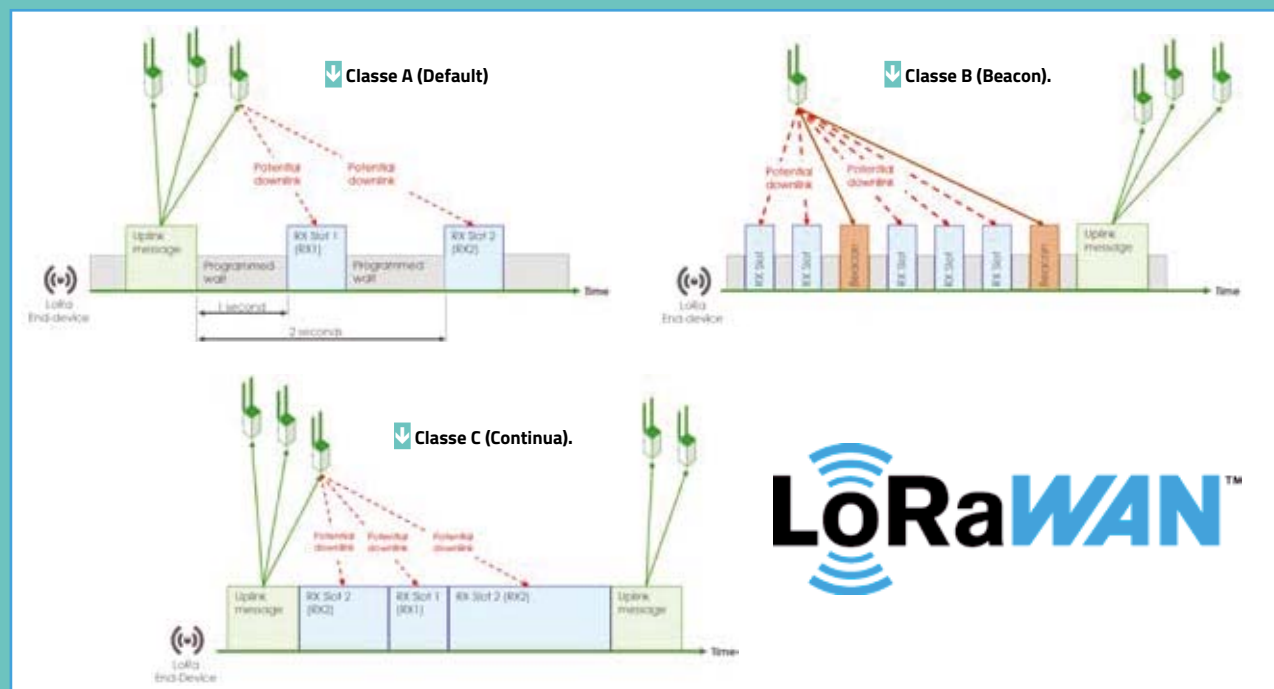
R: Il protocollo LoRaWAN è in grado di trasferire i dati del sensore o modificare il comportamento di un attuatore. LoRaWAN può essere utilizzato sia per la messaggistica uplink che per quella downlink e all'interno di LoRaWAN sono definite tre categorie di traffico, ovvero Classe A, Classe B e Classe C.

Classe A (Default)

La classe A è la classe predefinita, utilizzata per sensori o attuatori alimentati a batteria. Ogni messaggio di uplink è seguito da due slot di ricezione per i messaggi di downlink. Le trasmissioni in downlink vengono accodate in rete fino a quando il prossimo uplink crea due slot di ricezione. Tra i messaggi di uplink e le finestre di ricezione di downlink (RX1 e RX2), il dispositivo è in modalità di sospensione, con consumo di corrente praticamente nullo. Si noti che la rete non può inviare un downlink finché il dispositivo non si riattiva (attivato da un evento sul dispositivo come un timer o una soglia del sensore).

Classe B (Beacon)

I device di Classe B sono alimentati a batteria, in aggiunta alle due finestre della Classe A, la Classe B ha una finestra extra per i downlink, che viene aperta in momenti pianificati. Questa finestra è sincronizzata con il server attraverso un Beacon del gateway e viene utilizzata dal server per



sapere quando il device è in ascolto.

Classe C (continua)

I device Classe C sono alimentati dalla rete, le finestre di ricezione per i downlink sono quasi sempre aperte, l'unico momento in cui sono chiuse è durante la trasmissione.

Microbilancia a cristalli di quarzo

Su quale principio si basano le microbilance a cristalli di quarzo in grado di misurare valori di massa dell'ordine del microgrammo?

Sandro Martina • Genova

R: Le microbilance a cristalli di quarzo si basano sul meccanismo di trasduzione di un quarzo, il quale risiede nella sua capacità piezoelettrica, scoperta dai fratelli Curie nel 1880. È un elemento in cui le onde acustiche si propagano in una direzione perpendicolare alla superficie del cristallo. La frequenza di oscillazione è influenzata dallo spessore, dalla densità e dal tipo di taglio del quarzo e dalle proprietà fisiche del mezzo con cui il quarzo è posto in contatto. Proprio sfruttando questa caratteristica, nel 1959 Saurbrey ha dimostrato che la variazione della frequenza di oscillazione di un quarzo è collegata ad una massa solidamente adesiva alla sua superficie tramite una



relazione ben precisa.

In una fase liquida, il quarzo non è più solo un sensore di massa ma è influenzato anche e soprattutto dai parametri chimico-fisici della soluzione con cui è in contatto e la relazione che lega la sua variazione di frequenza cambia sensibilmente. Questa tecnica è relativamente semplice da usare e la strumentazione di base per utilizzarla è abbastanza economica. In Internet potrai trovare numerosi progetti, da quelli più semplici a quelli più complessi fino ai progetti completamente open source come OPENQCM.

La durata dell'impulso di clock

Nella quinta puntata del Corso dedicato alle Core Independent Peripherals di Microchip ho letto a pagina 118 che "...dato che il clock della nostra MCU è di 8 MHz e che ogni impulso di clock dura 25 nsec..." C'è forse qualcosa di sbagliato?

Giovanni Lupo • Napoli

R: Effettivamente nel testo pubblicato è saltato un "1" con l'impulso di clock che è evidentemente di 125 nsec anziché 25. La frase corretta è "...dato che il clock della nostra MCU è di 8 MHz e che ogni impulso di clock dura 125 nsec...". Ce ne scusiamo con i lettori.

Da Thunderbolt 3 a USB4

Oltre allo standard USB 3.2 ho letto che è in arrivo anche la specifica USB4, ancora più performante.

Mario Russo • Napoli

R: Appena annunciato, lo standard USB 3.2 sembra destinato a morte prematura per colpa (o merito) di Intel che già



nel 2017 aveva annunciato di voler "aprire" il suo Thunderbolt a tutti. Palla colta al balzo dall'USB-IF che sfrutterà le caratteristiche di Thunderbolt per realizzare una porta USB ancora più performante, ovvero quella USB4 che vedrà la luce entro il 2020 e che avrà una velocità di trasferimento di 40 Gbps; tra le altre caratteristiche l'adozione del formato Type-C, la possibilità di sfruttare la medesima porta anche come uscita video DisplayPort e l'erogazione di energia agli accessori esterni fino ad un massimo di 100 W. Per quanto riguarda la retrocompatibilità, la USB4 supporterà sia le periferiche Thunderbolt 3 sia quelle basate sui precedenti standard USB, incluso il 3.2 Gen 2x2.

Oltre 50 società partecipano attivamente alle fasi finali della revisione della bozza delle specifiche USB4, tra cui anche Apple. I primi dispositivi compatibili dovrebbero arrivare alla fine del 2020.

Web Forum e supporto tecnico

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

www.elettronicaonline.it/webforum



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

Progetti su misura!

Hai bisogno di una modifica ad un progetto pubblicato sulla rivista?

Manda una mail a
progettazione@futuragroupsrl.it
con una breve descrizione delle modifiche
che vorresti apportare per avere
un preventivo dettagliato.



› Microsoft guarda all'IoT

Nell'ottica di entrare anch'essa nel business dell'Internet of Things, il colosso di Redmond ha acquisito Express Logic, azienda specializzata in sistemi operativi real-time per dispositivi connessi IoT. Con questa mossa Microsoft punta a sfruttare sia le competenze, sia i prodotti (moltissimi dispositivi IoT integrano la tecnologia di Express Logic) dell'azienda californiana e in particolar modo a sfruttarne l'RTOS ThreadX, cui si affianca un file system e un pacchetto di risorse utili agli sviluppatori per creare interfacce grafiche. Insieme, questi prodotti compongono una piattaforma per l'embedded IoT. ThreadX è "complementare" ad Azure Sphere, la soluzione di Microsoft che consente di creare dispositivi IoT per i quali sono richieste connessioni al cloud molto sicure ed è rivolta a chi sviluppa



dispositivi con hardware molto limitati (per esempio quelli alimentati a batteria e dotati di meno di 64 kB di memoria Flash); infatti ThreadX fornisce un RTOS dotato di certificazioni aggiuntive e connettività preconfigurata con l'Azure IoT Hub, il che permette di integrare rapidamente e facilmente gli oggetti connessi nel cloud di Microsoft. Con l'acquisizione di ThreadX, Microsoft punta quindi a coinvolgere nel proprio cloud anche dispositivi IoT quali sensori che richiedono una modesta elaborazione di dati, che nell'immediato futuro diventeranno un'infinità,

piuttosto che orientare le proprie piattaforme esclusivamente all'edge-computing che serve grandi sistemi periferici.

<https://rtos.com>

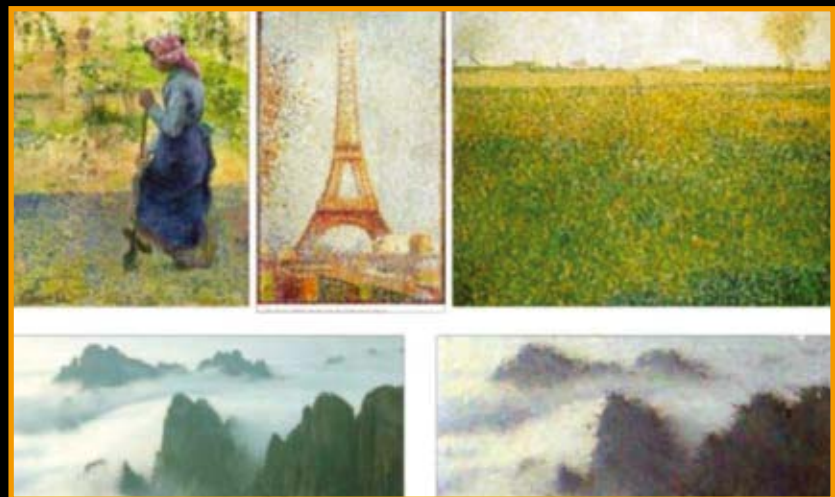
› L'Intelligenza Artificiale che dipinge

L'inarrestabile incedere dell'Intelligenza Artificiale non trascura l'arte e a dimostrarlo è una notizia che viene dall'Università del Maryland, dove è stato implementato un algoritmo chiamato PaintBot che impara a dipingere copie di quadri di grandi pittori. I ricercatori hanno addestrato l'algoritmo mediante un metodo di apprendimento basato sul rinforzo, ossia su sistemi in grado di correggere gli errori scartando i comportamenti infruttuosi e ripetendo quelli che hanno dato risultato positivo. Il tutto grazie a una complessa funzione matematica che calcola le differenze nella posizione degli elementi rappresentati e del colore, fra l'immagine prodotta dall'algoritmo e quella di riferimento. Gli autori dello studio hanno mostrato al sistema alcune immagini di riferimento, chiedendo di riprodurle come l'avrebbe fatto un pittore. Il sistema si

adattava gradualmente alla richiesta e produceva immagini digitali simili a quelle mostrate, analizzando la loro composizione per dimensioni, colore, posizione e densità della pennellata.

Il processo di apprendimento è durato circa sei ore, mentre per completare un quadro sono bastati 5 minuti!

<https://arxiv.org>



› ABB + Ericsson = Industrial IoT

Il colosso dell'automazione industriale e della robotica ABB ed Ericsson, tra i fondatori della comunicazione su rete cellulare, uniscono i loro sforzi e a suggellarlo hanno firmato un Memorandum of Understanding all'ultima edizione di Hannover Messe, per chiarire la loro visione comune sulla produzione flessibile integrata con l'automazione e la comunicazione wireless. L'intesa unirà le tecnologie per la connettività 5G e l'automazione industriale nel dare vita a sistemi Industrial IoT per l'Industria 4.0 e in generale alla Smart Factory, dove i macchinari saranno sempre più connessi e collaboranti, nell'ottica dell'ottimizzazione dei processi produttivi

e dell'efficienza, attraverso la manutenzione predittiva e la cooperazione tra macchine.

La leadership di ABB nella robotica industriale assistita dalla digitalizzazione e il lavoro pionieristico che Ericsson ha fatto da anni nell'ambito della connettività e nel 5G, aprirà nuove opportunità di miglioramento della produttività attraverso la creazione della fabbrica intelligente.

Ericsson e ABB già avevano avviato una stretta collaborazione nella ricerca riguardante l'applicazione della tecnologia 5G e dell'Industrial IoT all'automazione di fabbrica e alla robotica e con la sottoscrizione del MoU, rafforzano la partnership

› La Fabbrica del futuro secondo Bosch Rexroth

Nell'intento di perseguire la realizzazione della Fabbrica del Futuro, intesa come integrazione di processi produttivi a tutti i livelli, Bosch Rexroth sviluppa soluzioni per automazione e industrial IoT che passano dall'utilizzo del 5G come link wireless attraverso cui far viaggiare la mole di dati richiesta dalla comunicazione con i sensori e dal risultato dell'edge computing.

Un esempio di ciò Bosch Rexroth lo ha proposto con la stampante 3D completamente automatizzata e connessa in

5G, sviluppata in collaborazione con il produttore di stampanti 3D industriali BigRep, Nokia e Qualcomm.

Questa stampante 3D di grande formato (chiamata BigRep PRO) beneficia di soluzioni IIoT sviluppate da Bosch Rexroth e Bosch Connected Industry. Dispositivi IoT, software e sensori intelligenti monitorano in modo trasparente tutte le condizioni operative e ambientali e migliorano la qualità del processo di stampa, riducendo gli scarti e aumentando la qualità dei pezzi stampati.

Un apposito software ottimizza la durata utile della stampante, implementando il concetto di manutenzione predittiva: grazie alla capacità di rilevare tramite sensori le condizioni di usura ed eventuali errori, il software di monitoraggio interviene prima che l'usura delle parti causi un guasto tale da causare il fermo macchina. Il collegamento hardware 5G è supportato da un apposito software IoT Gateway integrato, che garantisce il collegamento della stampante con la rete Industrial IoT allo scopo di realizzare l'interconnessione con altre macchine e sistemi IT di livello superiore, creando linee di produzione connesse in rete.

I processi ad alta intensità di elaborazione, come il monitoraggio della qualità tramite visione artificiale possono essere trasferiti al cloud o ai sistemi Edge per la produzione, attraverso il 5G.

Tramite il Digital Service Assistant (DSA) e dispositivi mobili, gli addetti possono leggere la memoria dei difetti in modalità wireless, attivando la manutenzione predittiva.

Tra le novità Bosch anche CytroBox, una centralina idraulica connessa alla rete Online Diagnostics Network (ODiN) Rexroth.

www.boschrexroth.com



che punta ad accelerare il raggiungimento, per l'ecosistema industriale, dell'automazione flessibile, sbloccando nuove opportunità di business grazie alla combinazione tra 5G e Industry 4.0.

Tra i frutti della collaborazione tra le due aziende, lo sviluppo di una nuova generazione di tecnologie di fabbrica intelligente, dove Ericsson ha implementato un sistema di Smart Automation negli stabilimenti di produzione della ABB situati a Tallinn, in Estonia e ABB ha fornito una soluzione di cella robotica flessibile completamente automatizzata per assemblare gli apparati radio 5G per la Ericsson.

<https://new.abb.com>



› Il livello nei serbatoi si rileva con l'IoT



L'Internet of Things viene applicata ormai a ogni tipo di rilevamento, come testimonia l'azienda sudafricana PEPPERL+FUCHS, la quale ha recentemente reso disponibile un sensore di livello liquidi a ultrasuoni in grado di trasmettere in wireless i propri dati a una stazione remota connessa tramite una rete Industrial Internet of Things. Il sensore IoT Wilsen.sonic.level è

incluso in uno starter kit fornito da PEPPERL+FUCHS e comprendente il Wilsen.service e una piattaforma di archiviazione dati per facilitare la sperimentazione di questa nuova tecnologia.

Wilsen.sonic.level consente di implementare un sensore wireless ad ultrasuoni per monitorare i livelli di riempimento in contenitori mobili, serbatoi e

silos. L'utilizzo degli ultrasuoni per rilevare il livello dei liquidi consente una misurazione sicura e non affetta dal passaggio di luce, come potrebbe accadere utilizzando sensori a infrarossi, peraltro influenzabili da vapori ed esalazioni; inoltre rende i rilevatori adatti all'utilizzo con combustibili liquidi e in generale con liquidi infiammabili, grazie al fatto che non c'è alcun contatto in prossimità delle sostanze da monitorare. L'utilizzo degli ultrasuoni è anche ideale per le sostanze corrosive (acide e basiche) con le quali sarebbe difficile adottare i tradizionali sensori a galleggiante.

Inoltre, il dispositivo, che è alimentato a batteria, permette la geolocalizzazione del contenitore e trasmette i dati rilevati a un punto di raccolta sul web, tramite una connessione GSM o LoRaWan. Wilsen.service è il supporto che garantisce lo scambio sicuro dei dati e consente anche la gestione centralizzata di tutti i sensori IoT presenti sul campo, oltre a supportare la connessione con i sistemi software e le piattaforme per l'elaborazione dei dati dei clienti.

www.pepperl-fuchs.com

KIT COMPLETI PER VIDEOGIOCHI ARCADE CON RASPBERRY PI 3 B+

KIT RETROPIE PER VIDEOGIOCHI ARCADE CON CABINET

Postazione mono giocatore per videogiochi Arcade.
Il sistema utilizza il software RetroPie con il quale è possibile emulare vari tipi di console, tra cui Nintendo, Game Boy, MAME, Sega Master System, Amiga, Commodore, ecc. Il kit include: display LCD IPS a colori da 10" ad alta risoluzione, board Raspberry Pi 3 tipo B+, cabinet completo, SD-Card da 16GB con il software RetroPie precaricato. È dotata di 4 porte USB alle quali è possibile collegare dispositivi esterni, come ad esempio un GamePad e di porta ethernet per il collegamento a internet.



€ 279,00

Cod. ARCADE101



KIT RETROPIE PER VIDEOGIOCHI ARCADE

Costruisci una postazione per videogiochi Arcade come quelle degli anni '80 e '90! Grazie a questo kit potrai emulare vari tipi di console: Nintendo, Game Boy, MAME, Sega Master System, Amiga, Commodore...

€ 115,00

Cod. RETROPIEKIT_N



Attenzione: il cabinet non è compreso, ma i disegni per realizzarlo sono scaricabili dalla scheda online del prodotto.

KIT MINI RETROPIE PER VIDEOGIOCHI ARCADE

Il kit permette di realizzare un mini cabinet per videogiochi Arcade degli anni '80 e '90. Consente di emulare vari tipi di console: Nintendo, Game Boy, MAME, Sega Master System, Amiga, Commodore...
Dimensioni contenute: 250x250x250 mm.

€ 184,00

Cod. MINIRETROPIE



Attenzione: il case mini cabinet realizzato con stampante 3D è acquistabile separatamente (9 colori disponibili). Cod. CASEMINICABINET • € 189,00.

Prezzi IVA inclusa

**FUTURA
ELETTRONICA®**

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

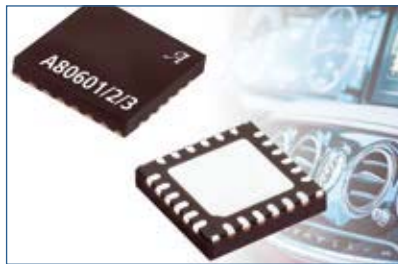
Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

LED driver A8060x con soppressione del rumore PWM

Allegro MicroSystems ha annunciato la famiglia A8060x, l'ultima generazione di driver avanzati per retroilluminazione a LED. La famiglia di dispositivi è progettata con un controllo Pre-Emptive Boost (PEB) innovativo e brevettato, che elimina il rumore generato.

Il controllo PEB nella famiglia A8060x riduce sostanzialmente l'ondulazione di Vout ed elimina il problema comune del rumore udibile generato dai condensatori di uscita in ceramica durante l'attenuazione PWM,



mentre richiede, in generale, una minore capacità di uscita. I sistemi possono otte-

nere un rapporto di contrasto luminosità a LED di 15.000:1 utilizzando la regolazione PWM a 200 Hz. Un rapporto più alto di 150.000:1 è possibile quando si utilizza una combinazione tra PWM e attenuazione analogica. L'ALT80600 e l'A80603 combinano un convertitore di commutazione con MOSFET integrato e quattro dissipatori di corrente, mentre l'A80601 e l'A80602 azionano un FET boost esterno per una maggiore potenza di uscita.

www.allegromicro.com

Cadence annuncia la prima IP di verifica per USB4 del settore

Cadence Design Systems ha annunciato la disponibilità della prima IP di verifica (VIP) del settore a supporto dello standard USB4, recentemente annunciato. Cadence VIP per USB4 consente agli ingegneri di sviluppare progetti System-on-Chip (SoC) conformi agli standard, completando la verifica funzionale del progetto con meno sforzo e ottenendo una maggiore certezza in merito al fatto che il SoC funzionerà come previsto.

USB Promoter Group ha annunciato il rilascio della pending release USB4, basata sul protocollo Intel Thunderbolt. Questa versione migliora il

trasferimento di video e dati, raddoppiando il throughput totale disponibile fino a 40 Gbps. Cadence ha sviluppato il nuovo USB4 VIP dopo oltre un anno di lavoro con i primi utilizzatori di questa nuova specifica; questa collaborazione ha portato alla disponibilità di un IP di verifica comprovato e maturo che riduce i rischi di progettazioni e accorcia i tempi di implementazione.

www.cadence.com



Moduli di potenza XHP 3 a 3,3 kV per inverter compatti e scalabili

Infineon Technologies estende il suo ampio portafoglio di dispositivi ad alta tensione con una nuova unità: XHP 3. Si tratta di una nuova flessibile piattaforma di moduli IGBT per applicazioni ad alta potenza nel campo di tensione da 3,3 kV fino a 6,5 kV. Il modulo consente progettazioni scalabili con la massima affidabilità e la massima densità di potenza. Grazie al suo design simmetrico con bassa induttanza parassita, offre un funzionamento di commutazione notevolmente migliorato. Per questo motivo, la piattaforma XHP 3 offre una soluzione per applicazioni impegnative di trazione in veicoli commerciali, edili e agri-

coli nonché azionamenti a media tensione.

Il pacchetto XHP 3 di Infineon presenta un fattore di forma compatto con 140 mm di lunghezza, 100 mm di larghezza e 40 mm di altezza. I primi moduli IGBT di questa nuova piattaforma ad alta potenza presentano una topologia a mezzo ponte con una tensione di blocco di 3,3 kV e una corrente nominale di 450 A.

www.infineon.com



Filtri di potenza AC-DC da CUI

Il Power Group di CUI ha annunciato l'introduzione di nuove linee di filtri di potenza AC-DC, espressamente ideati per completare la gamma di alimentatori AC-DC embedded a basso consumo della società. I dispositivi delle serie EMC-20 ed EMC-30 garantiscono un filtraggio del rumore di linea AC di 20 e 30 dB rispettivamente nella gamma di frequenza compresa tra 150 kHz

e 1 GHz. Disponibili in diverse configurazioni – per montaggio su scheda, su chassis e su guida DIN – tutti i modelli sono conformi alle specifiche degli



standard IEC/EN61000-4-5 per quanto riguarda i fenomeni di surge (sovratensioni transitorie) e permettono di ridurre le emissioni, sia condotte sia irradiate, semplificando in tal modo lo sviluppo di progetti che devono rispettare i limiti previsti dalla normativa CISPR22/EN55022 Class B.

Tra le altre caratteristiche dei nuovi filtri EMC AC-DC si possono annoverare l'ampio

range di tensioni di ingresso, compreso tra 85 e 305 Vac e la possibilità di accettare una corrente di ingresso nominale fino a 1,5 A. La tensione di isolamento pari a 2.000 Vac e l'esteso intervallo di temperatura operativa, variabile da -40 e +85 °C, conferiscono a questi filtri doti di robustezza e affidabilità che ne consentono l'uso in ambienti gravosi.

www.cui.com

Nuovi dispositivi in Carburo di Silicio (SiC) da Microchip

Microchip annuncia, tramite la sua controllata Microsemi, l'avvio in produzione di una famiglia di dispositivi SiC di potenza in grado di offrire comprovata robustezza ed i vantaggi prestazionali della tecnologia wide band-gap. Complementati dall'ampia gamma di microcontrollori

(MCU) di Microchip e dalle soluzioni analogiche, i dispositivi SiC si uniscono ad una crescente famiglia di affidabili prodotti SiC. I MOSFET SiC da 700 V e i Schottky Barrier Diode (SBD) da 700 V e 1200 V di Microchip vanno ad ampliare il portfolio esistente di moduli di potenza

SiC. Gli oltre 35 prodotti discreti che sono stati aggiunti al portfolio Microchip sono disponibili in volumi, supportati da completi servizi di sviluppo, strumenti e schemi di riferimento, e offrono eccezionale robustezza dimostrata da test rigorosi.

www.microchip.com



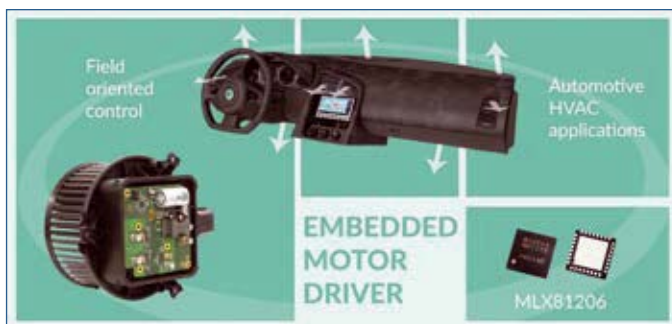
MLX81206, Melexis estende il proprio portafoglio di azionamenti per motori

Melexis presenta MLX81206, l'ultima aggiunta al proprio portafoglio di azionamenti incorporati di seconda generazione per applicazioni automotive. Questa soluzione su chip singolo con 64 KB di memoria Flash offre un grado superiore di integrazione e la possibilità di azionare motori BLDC da 100W a 1000W.

L'MLX81206 supporta un'interfaccia conforme agli standard LIN 1.x/LIN 2.x e SAE J602, gli standard di interfaccia di comunicazione digitale più am-

piamente utilizzati nel settore automotive, il che lo rende una soluzione pratica per il preazionamento di un'ampia gamma di motori BLDC ad alta potenza. Essendo in grado di azionare motori a 12V in configurazione con o senza sensore, l'MLX81206 costituisce l'opzione ideale per ventilatori HVAC silenziosi e ad alte prestazioni, ventole di raffreddamento dei motori e pompe nei veicoli convenzionali, ibridi e completamente elettrici.

www.melexis.com



Modulo combo Type 1LV con WiFi e Bluetooth

Murata, in collaborazione con Cypress Semiconductor Corp, ha sviluppato la soluzione Type 1LV (basata sul chipset CYW43012) che si propone come il modulo più piccolo e caratterizzato dai consumi più bassi al momento disponibile in grado di supportare le trasmissioni in modalità Wi-Fi e Bluetooth. L'utilizzo di questo modulo permette di aumentare la durata delle batterie in dispositivi indossabili, prodotti per la casa "intelligente" e apparecchiature audio portatili. Confrontando il nuovo modulo Type 1LV con il modulo Type 1DX (basato sul chipset CYW4343W), il consumo del nuovo dispositivo è inferiore di circa il 54% per un intervallo DTM1 (Delivery Traffic Indication Message) pari a 1 (DTMI-1), del 60% per un intervallo DTIM-3, fino al

50% per la ricezione (RX) a 2,4 GHz e del 28% per la trasmissione (TX) sempre a 2,4 GHz.

Basato sul chipset combo CYW43012 di Cypress, questo modulo Wi-Fi dual band di ridottissime dimensioni conforme alle specifiche IEEE 802.11 a/b/g/n/ (11ac Friendly) e Bluetooth 5.0 garantisce velocità di trasferimento dati fino a 78 Mbps (Wi-Fi) e 3 Mbps (Bluetooth). Il modulo supporta inoltre BLE 2Mbps, LE Secure Connections, LE Privacy 1.2 e LE Data Packet Length Extension.

www.murata.com



Micro con capacità di rilevamento di materiali solidi e liquidi



Renesas Electronics annuncia l'immediata disponibilità della "Material Detection Solution", una tecnica facile ed economica per rilevare liquidi o materiali senza l'utilizzo di sensori dedicati, misurando la capacità esistente tra due elettrodi connessi alla periferica Touch, integrata nel microcontrollore

Renesas RX130.

La sostituzione di sensoristica dedicata con questo tipo di approccio, contribuisce alla riduzione della lista dei componenti e permette un numero maggiore di punti di rilevamento, utilizzando un singolo dispositivo. Questo permette, ai produttori di apparecchiature industriali,

di automazione d'ufficio o elettrodomestici, di poter esplorare una nuova tecnica di rilevazione, in applicazioni dove è necessario il contenimento dei costi.

L'aumento o la diminuzione di polvere, liquido, o materiale, è in grado di cambiare il valore di capacità tra due elettrodi.

www.renesas.com

Piattaforma IoT multi-sensore BLE senza batterie da ON Semiconductor

ON Semiconductor proseguendo nella sua strategia finalizzata a fornire soluzioni IoT che non necessitano né di batterie né di manutenzione, ha annunciato l'introduzione della piattaforma multi-sensore RSL10 alimentata solamente con una cella fotovoltaica. Si tratta di una soluzione completa che supporta lo sviluppo di sensori IoT che utilizzano l'energia solare accumulata su base continua per acquisire e trasferire dati mediante la tecnologia Bluetooth Low Energy senza richiedere la presenza di batterie o altre forme di energia non rinnovabile.

La combinazione tra comunicazioni wireless a bassissimo consumo, una cella fotovoltaica di ridotte dimensioni e applicazioni di rilevamento con basso duty cycle, permette lo sviluppo e l'installazione di nodi sensori IoT che non

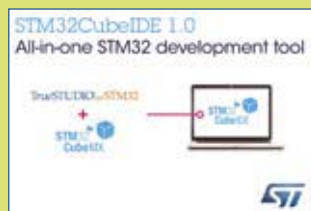


richiedono alcuna manutenzione. Per il funzionamento, la piattaforma multi-sensore a cella fotovoltaica RSL10 utilizza RSL10 SiP, una completa soluzione di tipo System-in-Package (SiP) che integra la radio RSL10, un'antenna e tutti i componenti passivi necessari. Questa innovativa piattaforma abbina RSL10 SiP con una cella fotovoltaica e un gran numero di sensori a basso consumo prodotti da Bosch, tra cui il sensore ambientale all-in-one.

www.onsemi.com

ST amplia l'ecosistema STM32Cube con l'IDE gratuito STM32CubeIDE

Prosegue l'impegno di STMicroelectronics per aumentare le funzionalità della famiglia di microcontrollori STM32 ad alta efficienza energetica con l'offerta nel nuovo IDE gratuito STM32CubeIDE, una soluzione all-in-one ora disponibile come parte dell'ecosistema software STM32Cube. Facile da utilizzare come ambiente di sviluppo integrato (IDE), STM32CubeIDE sfrutta l'acquisizione di ST del produttore di strumenti di sviluppo embedded Atollic nel 2017. È offerto con licenza open ed è in grado di aggiungere funzionalità dedicate specifiche per STM32, che possono semplificare e accelerare progetti embedded basati su questo micro. Tra queste, il potente strumento



STM32CubeMX per configurare il microcontrollore e gestire la costruzione del progetto. L'ecosistema STM32Cube è molto popolare tra gli sviluppatori, con una media di oltre 250.000 download all'anno. Combinando STM32CubeMX con STM32CubeIDE, si viene a creare un ambiente di sviluppo molto più potente dei tipici strumenti gratuiti offerti dai rivenditori di MCU.

www.st.com/stm32cubeide

Toshiba lancia un nuovo adattatore per moduli ottici unidirezionali

Toshiba Electronics Europe ha annunciato il lancio di un nuovo adattatore per moduli ottici unidirezionali. Esso è particolarmente utile nei sistemi industriali come i sistemi di accumulo dell'energia e gli inverter ad alta tensione.

Il nuovo TOCA1300 consente di collegare un modulo di trasmissione per fibra ottica (ad esempio il TOTX1350(F) di Toshiba) direttamente ad un modulo ricevitore compatibile (ad esempio il TORX1350A(F) o il TORX1355(F)), consentendo così la trasmissione dei dati su brevi distanze.

Il modulo TOCA1300 fornisce una soluzione semplice e fa-

cile all'uso per sistemi in cui è richiesta una distanza di isolamento ("distanza di separazione") superiore rispetto a quella supportata da un fotoaccoppiatore. È altrettanto utile laddove sia richiesto un accoppiamento più stretto con un cavo in fibra ottica. Il dispositivo è prodotto in volumi e le consegne sono già iniziate.

www.toshiba.semicon-storage.com



BD71847AMWV, power management IC di ROHM

ROHM annuncia oggi la disponibilità sul mercato del BD71847AMWV, un circuito altamente integrato per la gestione della potenza (PMIC), ottimizzato per la famiglia di processori i.MX 8M Mini prodotta da NXP Semiconductors (www.nxp.com), che presenta un'eccellente performance in fatto di computing, audio, video e grafica. Il BD71847AMWV è l'ultimo nato nel portafoglio prodotti di PMIC in continua espansione di ROHM, progettato appositamente per i processori i.MX di NXP. Insieme al BD71837MWV, introdotto sul mercato un anno fa per supportare i processori quadrupli e doppi ad elevata performance i.MX 8M, ROHM ora dispone di soluzioni di potenza per tutti i dispositivi della famiglia i.MX 8M.

I processori i.MX 8M Mini consentono ai clienti di in-

tegrare l'interfaccia vocale su dispositivi di streaming audio e video ad un costo minimo. Integrano fino a quattro Arm Cortex-A53 che funzionano fino a 1,8 GHz, insieme ad un Arm Cortex-M4 che funziona ad un massimo di 400 MHz per un basso consumo di energia in standby. Inoltre il supporto di elaborazione video a 1080p, la grafica 2D/3D, le funzioni audio avanzate e le diverse interfacce ad alta velocità fanno di i.MX 8M Mini il prodotto ideale per un'ampia gamma di applicazioni industriali e di consumo.

www.rohm.com/eu



Trasformiamo le tue idee in realtà!

Servizio taglio laser e incisione

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



Servizio stampa 3D

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

Informazioni dettagliate su
www.futurashop.it



➤ 24-27 GIUGNO 2019

LASER WORLD OF PHOTONICS 2019

MONACO DI BAVIERA • GERMANIA

LASER World of PHOTONICS è il principale appuntamento internazionale per l'industria del laser e della fotonica. Il salone si svolge in contemporanea con il World of Photonics Congress, il principale congresso di fotonica a livello europeo. Il programma del Congresso prevede diverse conferenze di organizzazioni internazionali. A completamento dell'offerta, Messe München propone presentazioni di applicazioni di fotonica ("Application Panels").

Nel 2017 il salone ha registrato la presenza record di 1.293 espositori da 42 Paesi. Complessivamente il Centro Espositivo di Monaco di Baviera ha accolto oltre 32.000 operatori qualificati da 90 nazioni. Nel 2017 il World of Photonics Congress ha accolto 3.500 partecipanti, proponendo oltre 3.000 presentazioni. Il salone LASER World of PHOTONICS viene organizzato dal 1973 con cadenza biennale da Messe München e quest'anno si svolgerà dal 24 al 27 giugno 2019.

I preparativi per il salone e il Congresso del prossimo giugno sono in pieno svolgimento. A che punto siamo? Lo

chiediamo a Katja Stolle, Project Manager di LASER World of PHOTONICS: *"Ci aspettiamo oltre 1.300 espositori da circa 45 Paesi: alzeremo ulteriormente l'asticella. Un settore in forte crescita è quello della visione e della sensoristica, per il quale voglio ricordare che, oltre alla conferenza "Optical Metrology" organizzata da SPIE, abbiamo programmato la conferenza "Imaging and Applied Optics" della OSA. Così nel 2019 avremo per la prima volta ben sette conferenze".*

Quali sono i temi di attualità di quest'anno? Chiediamo ancora. *"Visione e sensoristica sono le tecnologie che aprono la strada a processi di produzione con un alto grado di automazione e qualità nell'ambito dell'Industria 4.0, pertanto sono molto richieste in tutti i settori. Attualmente registriamo anche un forte impulso dall'industria automobilistica, spinta dalla mobilità elettrica e dai veicoli autonomi".*

La fotonica come tecnologia abilitante: parola chiave anche nelle bioscienze. E la fotonica quantistica? Lo chiediamo nuovamente a Katja Stolle: *"La tecnologia quantistica sarà in primo pia-*

no anche alla prossima edizione. La visione dell'ottica quantistica ha assunto in pochissimo tempo forme concrete. Il Prof. Anton Zeilinger presenterà il lavoro del suo team presso l'Istituto di Ottica e Informatica Quantistica dell'Accademia delle Scienze austriaca a Vienna. Ci saranno poi altre presentazioni su questo tema nella European Quantum Electronics Conference. Ma anche nei padiglioni la tecnologia quantistica avrà un ruolo importante. Molti espositori hanno già proposte concrete. Inoltre, il programma collaterale del salone prevede un doppio percorso di aggiornamento sull'applicazione delle tecnologie quantistiche nell'ambito della sensoristica e della simulazione e nell'ambito dell'imaging e della comunicazione. I temi spazieranno quindi dalla ricerca di base alle applicazioni concrete. Sono certa che anche in questo caso la fotonica ci mostrerà la strada verso il futuro. Chi vuole avere un'anteprima dettagliata del futuro non deve quindi mancare all'appuntamento in fiera a Monaco di Baviera dal 24 al 27 giugno 2019."

<https://world-of-photonics.com>

➤ 11 GIUGNO 2019

AUTOMOTIVE FORUM DI NATIONAL INSTRUMENTS BOLOGNA

Lavorare oggi nel settore automobilistico comporta dover affrontare sia le sfide dell'elettrificazione dei veicoli (EV) che quelle legate alla sicurezza della guida assistita ed autonoma (ADAS/AD).

L'edizione 2019 di NI Automotive Forum consentirà di scoprire come un approccio al test aperto e flessibile sia fondamentale per superare le sfide della continua innovazione tecnologica e come trasformare il test basato su piattaforma in un vantaggio competitivo.

Durante l'evento, esperti del settore illustreranno le soluzioni NI per superare le sfide legate al rapido e continuo evolversi dei requisiti di test e validazione, con dimostrazioni pratiche sulla validazione di sistemi ADAS ed EV.

Verranno illustrate inoltre le soluzioni per la gestione dei dati eterogenei e complessi della sensor fusion, la validazione sia in alta tensione che in simulazione di EV Powertrain, le metodologie sia in simulazione HIL che Record&Replay per test ADAS/AD e molto altro ancora.

Dal canto loro i Solution Partner di NI metteranno a disposizione la propria esperienza e conoscenze per soddisfare le esigenze di clienti in un settore in rapida evoluzione, grazie all'approccio basato su piattaforma aperta. L'evento sarà anche un'occasione per

incontrare colleghi e per visitare le soluzioni proposte da National Instruments e dai suoi Partner nell'apposita area espositiva.

A Bologna San Lazzaro l'11 Giugno 2019 presso l'Unaway Congress Hotel.

www.ni.com



SMART VISION FORUM

Nasce dalla collaborazione tra AldAM e ANIE Automazione, Smart Vision Forum, la prima mostra-convegno dedicata ai sistemi di visione industriale. L'appuntamento organizzato da Messe Frankfurt Italia si terrà il **25 giugno nella Sala Maggiore di Bologna Congressi**.

Produttori, distributori e integratori di sistemi si confronteranno con il mondo accademico e industriale nel corso del programma convegnistico, articolato in una plenaria di apertura e in due sessioni parallele su tecnologie e applicazioni.

Nell'area espositiva le aziende presenteranno le soluzioni più innovative, al passo con la rapida evoluzione di questi

dispositivi. Si parlerà di tutti gli ambiti in cui il ruolo della smart vision diventerà sempre più strategico.

Hanno già aderito: Advanced Technologies, Alkeria, Balluff, Beckhoff, B&R, Datalogic, ifm electronic, iMAGE S, Omron, Keyence Italia, SICK, VEA, Video Systems, VISION, Visionlink, Wenglor.

Smart Vision Forum si pone l'obiettivo di

fare il punto della situazione sulle opportunità a disposizione della fabbrica intelligente per il raggiungimento di risultati tangibili quali l'affidabilità dei prodotti, la velocità delle procedure di controllo, il miglioramento della qualità e di conseguenza la fidelizzazione della clientela.

Michele Viscardi, Presidente AldAM ha dichiarato: "AldAM è particolarmente orgogliosa di essere partner - insieme ad ANIE Automazione e a Messe Frankfurt Italia - di un'iniziativa che offre importanti opportunità di confronto e crescita per l'intera filiera: produttori, integratori e utilizzatori."

<https://smartvisionforum.it/>



FORTTRONIC 2019: POWER E AUTOMOTIVE IN PRIMO PIANO

L'elettronica di potenza sarà al centro del prossimo **Fortronic Forum**, l'evento tecnologico di Assodel (Associazione Distretti Elettronica - Italia) in programma il 19 e 20 giugno 2019 a Modena.

Tra i mercati stimati in crescita, quello del **power electronics** resta uno dei più interessanti sia a livello internazionale (si parla di un +7% l'anno) sia a livello italiano, dove si osserva un dinamismo e una vivacità ancora più accentuati nella miriade di applicazioni e di PMI presenti. E non è un caso che l'Emilia-Romagna, territorio particolarmente vocato all'innovazione - con 14 centri specializzati, 10 tecnopoli e 82 laboratori di ricerca industriale - sia stata designata sede di questo evento a elevato contenuto tecnologico. Accanto al tradizionale focus sul Power, il Fortronic Forum 2019 punterà anche su un'altra area strategica per il mercato dell'elettronica: l'**automotive**. Le ultime edizioni di elettronica di Monaco e del CES di Las Vegas ci hanno mostrato come l'auto sia oggi fulcro di sperimentazioni e innovazioni tecnologiche di ogni genere e tipo. Dalle soluzioni di infotainment ai sistemi di sicurezza, dagli ADAS ai sensori, dai motori elettrici ai sistemi per la guida autonoma. L'argomento è trainante per tutto il comparto dell'elettronica e si sposa perfettamente con la scelta di Modena, situata nel cuore della Motor Valley.

Tre eventi in uno

Da oltre 20 anni, i Fortronic Forum sono giornate di networking e approfondimento su tematiche verticali pensate per rispondere alle esigenze di aggiornamento tecnologico di progettisti e buyer



attraverso contenuti autorevoli e incontri snelli, rapidi e incisivi.

L'edizione 2019 di Fortronic propone tre eventi in uno:

- **Strategic Innovation Summit** (19 giugno), un momento di ispirazione sull'innovazione tecnologica dedicato a manager e imprenditori e focalizzato sul mercato dei veicoli elettrici. L'evento vedrà la presentazione di alcuni casi di successo italiani andando ad affrontare anche alcune criticità del settore come le infrastrutture di ricarica, le batterie, i costi e la compatibilità elettromagnetica.
- **Fortronic Power & more** (19 e 20 giugno), il punto di riferimento per il mercato dell'elettronica di potenza... ma non solo (come detto sarà presente anche un focus sull'automotive). Tra area espositiva, conferenze, workshop e dimostrazioni dalle aziende, il visitatore avrà una visione a 360° di

tutta la filiera e potrà scegliere le soluzioni e le tecnologie migliori a supporto dei propri progetti innovativi. Tra i contenuti, si parlerà di magnetici, di tecnologie SiC e GaN, di compatibilità elettromagnetica, di supercap, di power digitale, di batterie e di Power Management IC.

- **Assodel Gala & Award** (19 giugno), la tradizionale cena dell'associazione, momento di networking per eccellenza del comparto. Nel corso della serata verranno assegnati i premi ai best manufacturer 2019 e ai clienti più innovativi (Industry Award).

Il Fortronic Forum 2019 presenterà alcune novità importanti con l'introduzione di un'area dimostrativa dedicata alla trazione elettrica e un Training Lab ("hands on") in cui i progettisti potranno toccare con mano prodotti e soluzioni a carattere innovativo.

<https://fortronic.it>

MOSTRE MERCATO

LOCRI (RC)

MOSTRA MERCATO DEL RADIOAMATORE, DELL'ELETTRONICA E DEL COMPUTER

Istituto di I° Grado F. Maresca

Organizzazione: G.C.S.

Telefono: 3382704739

www.cisarlocri.net - gallcosimo@tiscali.it

20 e 21 Luglio 2019

CEREA (VR)

FIERA DELL'ELETTRONICA DI CEREA

AreaExpo - Via Oberdan - Cerea (VR)

Organizzazione: fieraeelettronica.it

Telefono: 0308376078

www.fieraeelettronica.it

info@fieraeelettronica.it

31 Agosto e 01 Settembre 2019

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

www.elettronica.in.it/calendario-mostre-mercato

L'elenco completo dei più importanti eventi nazionali e internazionali di elettronica, sicurezza e fonti rinnovabili è disponibile sul sito

www.elettronica.in.it/blog/category/eventi/eventi_e_manifestazioni



ABBONAMENTO
ANNUALE



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

10
NUMERI

A SOLI

€45

ANZICHÉ € 60,00

COME ABBONARSI

→ on-line

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

→ mail

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

→ posta

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

FUTURA GROUP srl
Via Adige 11
21013 Gallarate (VA)

→ telefono

telefonando a
+39-0331-752668

Visita: www.elettronica.in.it/abbonamenti

VANTAGGI PER GLI ABBONATI

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

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

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

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

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



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



DISCOUNT CARD Futura Elettronica

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



**Fishino - Arduino e l'Internet delle
Cose in un'unica innovativa scheda**



Randa - Raspberry Pi + Arduino



**ARDUINO UNO - Programmazione
avanzata e Librerie di sistema**



Impara l'elettronica sperimentando



Primi passi in elettronica



**Arduino e le tecniche
di programmazione
dei microcontrollori ATMEL**



**Raspberry Pi - il mio primo
Linux Embedded**



Alla scoperta di Arduino YÚN



Programmiamo con i PIC



L'ABC di Arduino



Stampiamo in 3D

Resto in attesa del primo numero.

Cognome:

Nome:

Via:

N°:

CAP:

Città:

Provincia:

Telefono:

e-mail:

Data:

Firma:

Resto in attesa di vostre disposizioni per il pagamento.

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



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



Fishino

Design With Simplicity

I TUOI PROGETTI DIVENTANO WIRELESS

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

Fishino UNO



cod. FISHINO UNO
€ 36,00



- Controller: ATmega328
- Alimentazione:
 - da 7 a 12 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Compatibile al 100% con Arduino UNO
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Modulo RTC con batteria di mantenimento
- Sezione di alimentazione a 3,3V potenziata
- Compatibile con shield e schede millefori
- Peso 25 g
- Dimensioni 76,5 x 53,5 x 14 mm

- Controller: ATmega2560
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino MEGA
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Modulo RTC con batteria di mantenimento
- Stadio di alimentazione switching (5Vdc e 3,3Vdc)
- Compatibile con shield e schede millefori
- Peso 37 g
- Dimensioni 101,5 x 53,5 x 15 mm



Fishino MEGA



cod. FISHINOMEGA
€ 49,90

Fishino32



cod. FISHINO32
€ 59,00



- Controller: PIC Microchip a 32 bit
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 3,5 a 20 Vdc (tramite plug o sull'ingresso Vin)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile con Arduino UNO
- Modulo WiFi
- Modulo RTC con batteria di mantenimento
- Clock 120 MHz, riducibile via software
- Interfaccia per scheda microSD
- Codec audio stereo
- Peso 29 g
- Dimensioni 76,5 x 53,5 x 14 mm

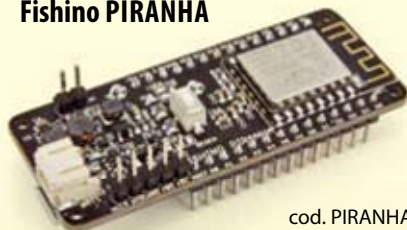
Fishino GUPPY



cod. GUPPY
€ 33,90

- Controller: ATmega328
- Alimentazione:
 - 3,6 Vdc (tramite batteria esterna al litio)
 - da 6,5 a 20 Vdc (tramite PIN VIN)
 - 5 Vdc (tramite porta USB)
- Circuito di ricarica per batteria al litio
- Compatibile al 100% con Arduino NANO
- Modulo WiFi
- Interfaccia per scheda MicroSD
- Peso 10 g
- Dimensioni 75 x 20 x 18,5 mm

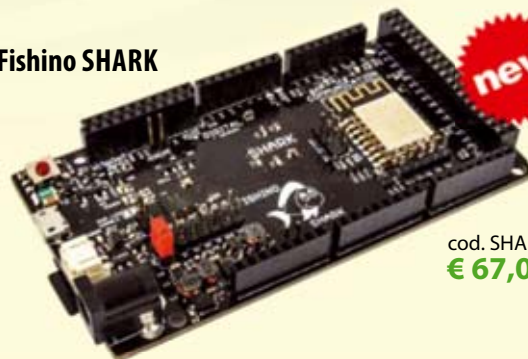
Fishino PIRANHA



cod. PIRANHA
€ 44,90

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

Fishino SHARK



cod. SHARK
€ 67,00

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

Utilizza subito Fishino UNO!

Set contenente il necessario per realizzare applicazioni con Fishino UNO.

cod. FISHINOSTARTER
€ 65,00



FISHINO IR REMOTE



di **CARLO PALUMBO**

Uniamo la potenza di Fishino UNO alla comodità degli assistenti vocali Google Home e potremo dimenticarci del telecomando e gestire vari dispositivi con delle semplici frasi.

C

on questo progetto andremo a sfruttare i nuovi ecosistemi tecnologici che stanno popolando le nostre case: gli assistenti vocali. In particolar modo, vedremo l'intero flusso di progettazione di un'azione personalizzata. Purtroppo, tutt'ora alcune funzioni sono ancora un po' limitate: ad esempio, per accendere o spegnere il nostro televisore avremo bisogno di un dispositivo esterno come Chromecast e di un televisore provvisto di tecnologia CEC (come Anynet+ HDMI-CEC), oppure se vogliamo cambiare canale o alzare/abbassare il volume del nostro televisore è impossibile farlo se non acquistando device che non sempre riescono ad essere al 100% compatibili. Vedremo, quindi, come si possono estendere le capacità di controllo della



nostra casa tramite l'utilizzo di una scheda come Fishino UNO ed un semplice LED a infrarossi, aggiungendo anche quel tocco di personalità che la renderà unica!

SCHEMA A BLOCCHI

Per poter realizzare questo progetto, avremo bisogno innanzitutto di "leggere" il nostro telecomando. Per farlo, utilizzeremo un ricevitore di segnali infrarossi che collegheremo a Fishino. Successivamente, con la medesima scheda, sostituiremo il ricevitore con un emettitore che si occuperà della comunicazione con il televisore. In **Fig. 1** è schematizzato il funzionamento della comunicazione a infrarossi, che verrà approfondito nei prossimi paragrafi. Andiamo ora a vedere quale sarà il flusso dati del nostro progetto: Google Assistant una volta ricevuta una frase di trigger da noi impostata invierà il messaggio al cloud dell'app IFTTT che lo elaborerà ed eseguirà il comando. Quest'ultimo provvederà all'invio di una richiesta GET sull'IP pubblico del nostro modem che a sua volta lo reindirizzerà alla scheda Fishino UNO. Infine, verrà inviato il segnale tramite il LED a infrarossi collegato. Può sembrare molto complesso, ma in realtà è facilissimo da realizzare. Inoltre, il sistema elabora molto velocemente tutto il flusso dati senza soffrire di particolari latenze. La **Fig. 2** rappresenta schematicamente quello che è stato appena spiegato.

FUNZIONAMENTO DEL TRASMETTITORE

Per l'invio del nostro segnale infrarosso utilizzeremo un semplice LED (reperibile su www.futurashop.it con il codice 1418-LED51R) capace di emettere, appunto, una radiazione luminosa nello spettro dell'infrarosso. Questa

radiazione non è visibile all'occhio umano, ma se ad esempio provassimo a premere un pulsante sul nostro telecomando e ad inquadrare il LED con la fotocamera del nostro cellulare, noteremo che il componente, di fatto, si accende e si spegne molto rapidamente. In particolare, ogni telecomando utilizza un suo protocollo e una sua frequenza per l'invio del segnale, sebbene si tenda a standardizzare i prodotti sulla frequenza di 38kHz. La scelta di questa frequenza non è casuale, infatti è molto difficile trovare dispositivi o disturbi naturali a quella frequenza e ciò rende il segnale meno soggetto a rumore. Per quanto riguarda il protocollo, invece, rimane diverso per ogni casa produttrice. Ciascuno può differire dagli altri per quantità di bit utilizzati per trasmettere l'informazione, banda passante, peso dei bit e distanza della modulazione tra un treno d'impulsi e l'altro. Per fortuna la libreria che utilizzeremo implementa i protocolli più usati, semplificandoci di molto il lavoro.

FUNZIONAMENTO DELLA RICEVENTE

Per ricevere un segnale infrarosso useremo un semplice ricevitore a fotodiodo IR calibrato sui 38kHz mediante un filtro, come ad esempio il TSOP58238 o il TSOP38238.

Con riferimento alla **Fig. 3** andiamo ad analizzare cosa è contenuto in questo componente.

- **Amplificatore e limitatore:** questi due blocchi vanno a costituire quello che viene chiamato AGC (*automatic gain control*) che serve ad evitare distorsioni in ampiezza del segnale, mantenendo quindi costante il livello di energia. A volte, per diminuire i disturbi dell'alimentazione, è consigliato aggiungere anche un condensatore e una resistenza nelle vicinanze del componente

Fig. 1
Schematizzazione della comunicazione IR.

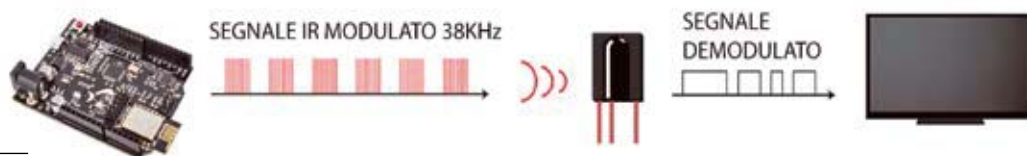
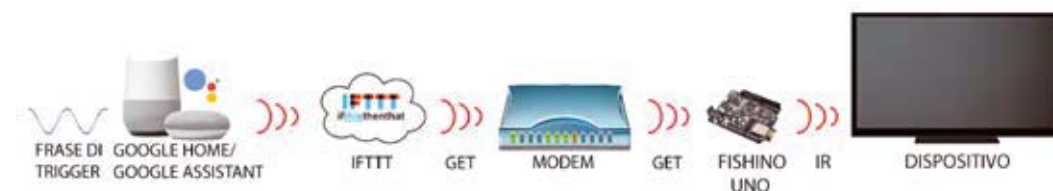


Fig. 2
Elaborazione del flusso dati.



- ricevitore, i cui valori e collegamenti sono indicati nel datasheet dello stesso.
- **Filtro passa-banda:** qui il segnale viene selezionato in base alla frequenza passante del filtro, nel nostro caso 38kHz.
 - **Demodulatore, integratore e comparatore:** questi tre blocchi servono per intercettare le informazioni presenti nel segnale e, in caso di verifica positiva, portare l'uscita ad un livello basso.

SCHEMA ELETTRICO PER LA RICEZIONE DEL SEGNALE

Questa configurazione di Fishino UNO servirà unicamente per poter memorizzare i vari tasti del nostro telecomando ma non verrà utilizzata per comandare il nostro dispositivo. I collegamenti sono immediati: basterà seguire le indicazioni del datasheet del componente. Nel nostro caso utilizzeremo un TSOP58238, per il cui basterà seguire lo schema in **Fig. 4**.

SCHEMA ELETTRICO PER L'INVIO DEL SEGNALE

Una volta "campionata" la stringa di ciascun tasto bisognerà realizzare l'elettronica che costituirà il nostro trasmettitore IR, proposta dalla **Fig. 5** per quando riguarda l'utilizzo di Fishino UNO. Potete comunque utilizzare una scheda differente, ricordando che il piedino con cui pilotare il LED infrarosso differisce, secondo quanto esposto nella **Tabella 1**. Per quanto riguarda l'alimentazione, notate che la soluzione più pratica sarebbe alimentare la scheda tramite un'uscita USB del proprio televisore. In questo caso, però, se spegniamo il TV è probabile che l'alimentazione venga a mancare, impedendo l'utilizzo del telecomando per accenderlo, dato che in molti televisori, allo spegnimento si disattiva l'USB pur mantenendo la spina nella presa. Un'alternativa più semplice è alimentare la scheda con un power bank o una batteria LiPo. Inoltre, se si è in possesso di una Fishino UNO REV2, l'opzione migliore è utilizzare una LiPo ed aggiungere allo sketch qualche riga di codice per implementare la funzione LOW POWER prolungando la durata della batteria.

LA COMUNICAZIONE A INFRAROSSI

Soffermiamoci, ora, sulla modalità di comunicazione mediante portante a raggi infrarossi. Premettiamo che questo paragrafo è finalizzato a mostrarvi didatticamente come avviene una trasmissione IR,

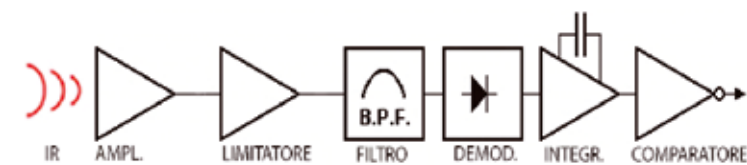


Fig. 3
Schema a blocchi del sensore IR.

non spaventatevi, la libreria IRremote farà tutto questo lavoro per noi. Prendiamo ad esempio il protocollo di comunicazione di un telecomando con codifica Samsung che utilizza 32bit per l'invio delle informazioni. Per poter analizzare il segnale inviato basta collegare la sonda di un oscilloscopio al pin positivo del

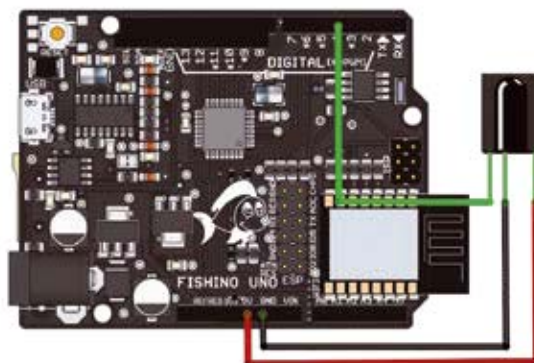


Fig. 4
Fishino e il ricevitore IR.



Fig. 5
Fishino collegato al LED IR.

SCHEDA	PIN
FISHINO UNO/GUPPY	3
FISHINO MEGA	5
ARDUINO UNO/MINI/NANO	3
ARDUINO MEGA	5
SPARKFUN PRO MICRO	5
NODEMCU	2

Tabella 1
Schede utilizzabili e piedino da usare.

Fig. 6
Gli impulsi che rappresentano la stringa.

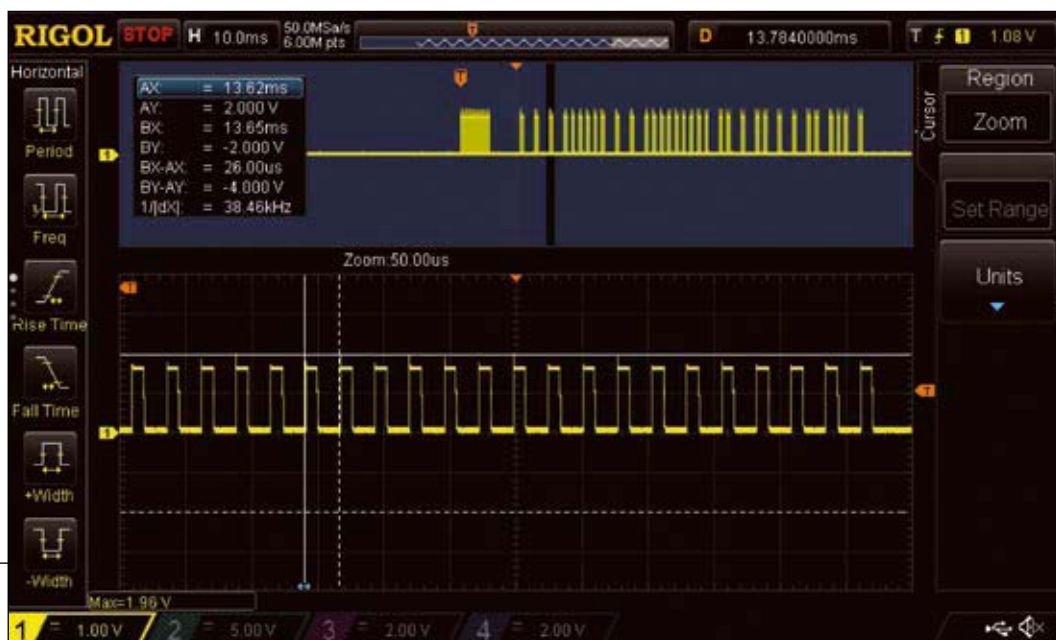
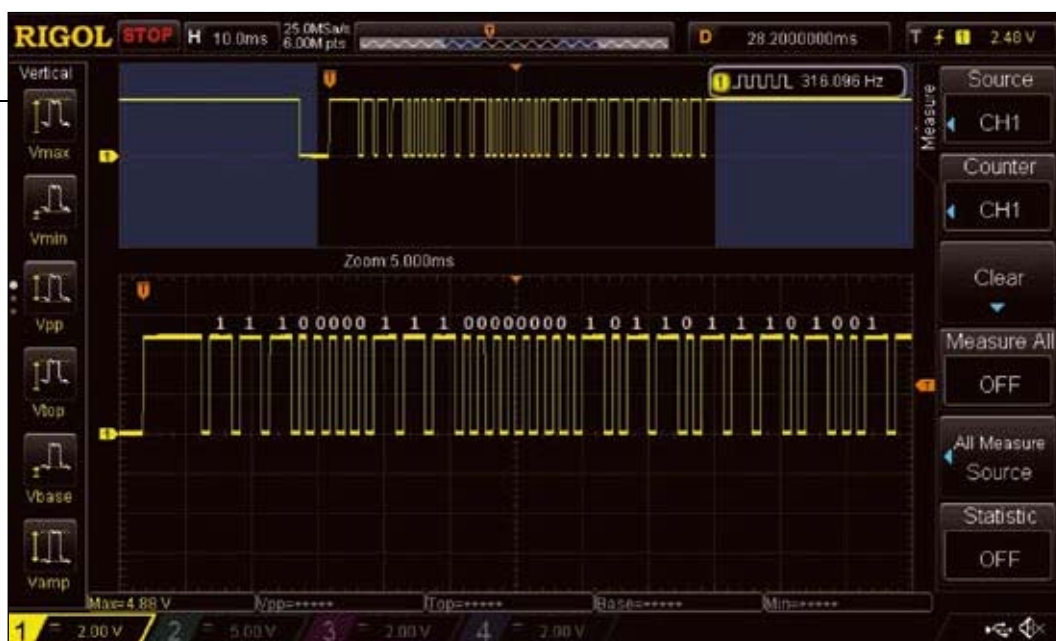


Fig. 7
Il protocollo Samsung.



LED a infrarossi pilotato da Fishino UNO: inviando una stringa in esadecimale, ad esempio E0E016E9, e codificandola rispettando il protocollo Samsung (per farlo bisogna utilizzare la libreria **IRremote2.h**), otterremo quanto mostrato in **Fig. 6**. Come vedete, nella parte superiore può sembrare che gli impulsi siano singoli, ma ingrandendo l'immagine possiamo vedere come ognuno di essi sia composto da una serie di piccoli impulsi molto veloci. In particolare, possiamo anche notare che la loro frequenza è proprio 38kHz! Viene quindi utilizzata la tecnica

PWM nel momento dell'invio del segnale. È importante evidenziare come nella parte superiore dell'immagine alcuni impulsi siano ad una distanza diversa da altri e questo servirà al ricevitore per andare a costruire l'informazione contenuta nel segnale inviato. Il segnale in uscita dal ricevitore sarà quindi amplificato e demodulato. Utilizzando la scheda Fishino UNO connessa a un ricevitore IR e collegando una sonda dell'oscilloscopio al pin preposto alla trasmissione del segnale ricevuto, inviamo la stessa

stringa esadecimale generata in precedenza. Sullo strumento avremo un risultato come quello in **Fig. 7**. Ciò che visualizziamo è una serie di impulsi a frequenza più bassa, che rappresentano la stringa. In particolare, il segnale inizialmente è a livello alto (contrariamente a prima) per poi andare a zero per 8ms e tornare alto per altri 4ms. Ciò indica l'inizio della trasmissione. Successivamente notiamo la sequenza decodificata dove le pulsazioni di durata 1ms indicano uno "0" e quelle di 2ms, "1". Il risultato sarà il codice binario mostrato nella figura che in esadecimale è proprio E0E016E9!

USO DELLA LIBRERIA IRREMOTE IN INVIO E RICEZIONE

La libreria su cui si basa l'intero progetto è *IRremote* e, come già accennato, ci permetterà di decodificare e in seguito ricodificare un pacchetto dati trasmesso tramite infrarosso. Senza questa

libreria, il lavoro sarebbe stato molto più complesso. Procediamo dividendo in due parti il ruolo della libreria: ricezione e invio del segnale.

Per quanto riguarda la ricezione faremo uso dello schema di **Fig. 4** e caricheremo sulla scheda lo sketch *ArduinoRemoteReceiver.ino* (**Listato 1**) scaricabile dal nostro sito web (elettronica.in.it). Analizziamo ora il codice; la prima cosa che notiamo è l'oggetto `irrecv(RECV_PIN)`, che indica il pin della scheda a cui è connesso il pin del LED ricevitore. Subito dopo abbiamo l'oggetto `results`, in cui verrà memorizzata l'informazione catturata per poter essere successivamente utilizzata all'interno del nostro sketch. Carichiamo, quindi, lo sketch, apriamo il monitor seriale e premiamo un tasto del telecomando. Avremo una situazione simile a quella mostrata in **Fig. 8**.

Purtroppo, nel caso in cui il protocollo non sia NEC, RC5, RC6 o Sony dovremo ricavare il segnale



Fig. 8
I dati corrispondenti alla trasmissione IR su Serial Monitor.

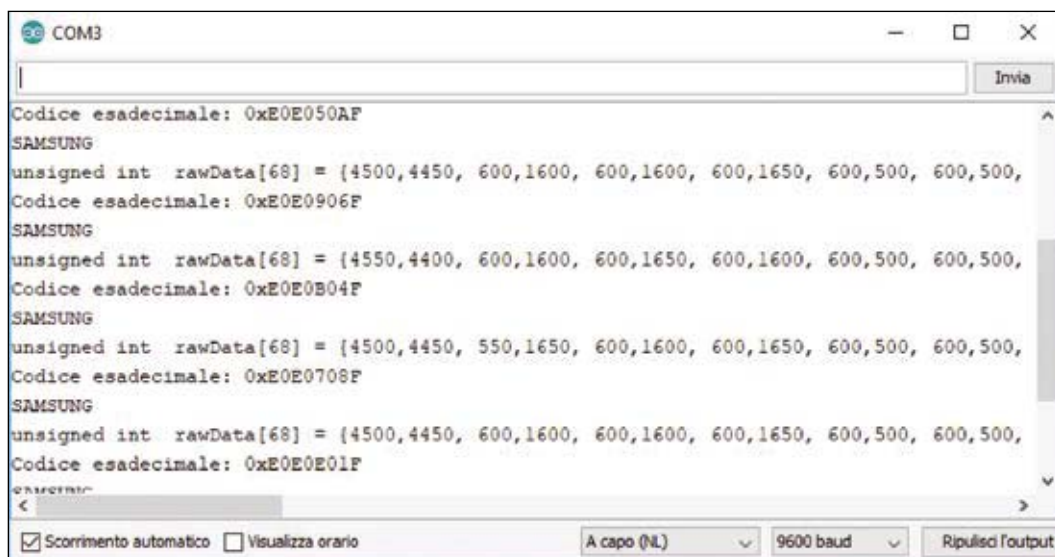


Fig. 9
Il codice RAW ottenuto.



Listato 1

```
//ArduinoRemoteReceiver by Carlo Palumbo

#include <IRremote.h>

//Pin dove è connessa l'uscita del nostro led ricevitore
const int RECV_PIN = 5;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup(){
  Serial.begin(9600);
  irrecv.enableIRIn();
  //Accende il led 13 ogni volta che riceve un segnale
  irrecv.blink13(false);
}

void dumpCode(decode_results *results){

  //Dichiarazione dell'array
  Serial.print("unsigned int ");          // Tipo variabile
  Serial.print("rawData[");              // Nome dell'array
  Serial.print(results->rawlen, DEC);      // Dimensione dell'array
  Serial.print("] = {");                  // Inizio dichiarazione dell'array

  //Dump dei dati
  for (int i = 1; i < results->rawlen; i++) {
    Serial.print(results->rawbuf[i] * USECPERTICK, DEC);
    if ( i < results->rawlen-1 ) Serial.print(","); // ',' not needed on last one
    if (!(i & 1)) Serial.print(" ");
  }

  //Fine dichiarazione dell'array
  Serial.print("};"); //

  //Nuova linea
  Serial.println("");

  //Dump del codice raw in caso di decodifica UNKNOWN
  if (results->decode_type != UNKNOWN) {

    //Se la codifica è Panasonic, avremo bisogno anche di un indirizzo
    if (results->decode_type == PANASONIC) {
      Serial.print("unsigned int addr = 0x");
      Serial.print(results->address, HEX);
      Serial.println("");
    }
  }
}

void loop(){
  //Se riceve un segnale, lo mostra in esadecimale e aggiunge il tipo di codifica trovata. Se necessario mostra anche la codifica RAW
  if (irrecv.decode(&results)){
    Serial.print("Codice esadecimale: 0x");
    Serial.println(results.value, HEX);
    switch (results.decode_type){
      case NEC: Serial.println("NEC"); break ;
      case SONY: Serial.println("SONY"); break ;
      case RC5: Serial.println("RC5"); break ;
      case RC6: Serial.println("RC6"); break ;
      case DISH: Serial.println("DISH"); dumpCode(&results); break ;
      case SHARP: Serial.println("SHARP"); dumpCode(&results); break ;
      case JVC: Serial.println("JVC"); dumpCode(&results); break ;
      case SANYO: Serial.println("SANYO"); dumpCode(&results); break ;
      case MITSUBISHI: Serial.println("MITSUBISHI"); dumpCode(&results); break ;
      case SAMSUNG: Serial.println("SAMSUNG"); dumpCode(&results); break ;
      case LG: Serial.println("LG"); dumpCode(&results); break ;
      case WHYNTER: Serial.println("WHYNTER"); dumpCode(&results); break ;
      case AIWA_RC_T501: Serial.println("AIWA_RC_T501"); dumpCode(&results); break ;
      case PANASONIC: Serial.println("PANASONIC"); dumpCode(&results); break ;
      case DENON: Serial.println("DENON"); dumpCode(&results); break ;
      default:
        case UNKNOWN: Serial.println("UNKNOWN"); dumpCode(&results); break;
    }
    //Attende un nuovo segnale da decodificare
    irrecv.resume();
  }
}
```


attraverso il codice RAW. In realtà esistono molte librerie che implementano singoli protocolli, come ad esempio la *IRremote2* per Samsung, quindi cercando un po' in rete sicuramente troverete la libreria adatta al vostro TV.

Ad ogni modo, nello sketch si avrà già integrata la funzione *dumpCode* che ci permetterà di ottenere il codice RAW del nostro segnale nell'eventualità di codifiche diverse; in questo modo avremo qualcosa di simile alla **Fig. 9**. In entrambi i casi annotiamo il codice di accensione/spegnimento del dispositivo per poterlo utilizzare nella fase successiva. Il test d'invio del segnale è ancora più immediato. Ci basterà, infatti, collegare la scheda al LED tramite una resistenza da 100 ohm, come già mostrato nella **Fig. 5**. Come possiamo notare dal **Listato 2**, che rappresenta lo sketch *ArduinoRemoteSender.ino*, le prime righe di codice sono occupate dalle variabili che andranno sostituite una con il valore annotato in precedenza in esadecimale o RAW e l'altra inizializzata a zero come suggerito nel commento soprastante.

Subito dopo abbiamo la variabile *isRaw* che ci permette di scegliere quale tipo di codifica si desidera utilizzare, per poi continuare con l'oggetto *irsend* che viene utilizzato per gestire il segnale che vogliamo in uscita. È evidente che in questo caso non è possibile selezionare alcun pin. Infatti, la libreria per l'invio dei dati si serve del timer interno della scheda per generare gli impulsi alla frequenza prestabilita. Ecco il motivo per cui è importante rispettare la **Tabella 1** mostrata in precedenza. Infine, le righe all'interno della funzione *loop* servono ad inviare il segnale di accensione/spegnimento al dispositivo se scriviamo il carattere 'o' sul monitor seriale. Anche qui qualche precisazione: se inviamo un segnale codificato Raw mediante l'istruzione *irsend.sendRaw(rawbuf, rawlen, frequency)*, stiamo attenti ad inserire la giusta lunghezza del nostro array e la frequenza che vogliamo utilizzare, comunemente è 68 per la prima e 38 per la seconda.

Se abbiamo invece la fortuna di poter sfruttare le funzioni della libreria, ricordiamoci di utilizzare il comando relativo (*sendNEC(IRcode, numBits)*, *sendRC5(IRcode, numBits)*, *sendRC6(IRcode, numBits)*, *sendSony(IRcode, numBits)*) e inserire il giusto numero di bit come riportato in **Tabella 2**.

Una volta caricato lo sketch e inviato alla scheda il carattere 'o', il nostro dispositivo (in questo caso un televisore) si accenderà.

Appurato, quindi, che il sistema funziona, si può tornare allo schema utilizzato per la ri-

cezione, caricare nuovamente lo sketch *ArduinoRemoteReceiver.ino* e mappare tutti i pulsanti che siamo interessati a codificare, annotandone i codici in un file di testo.

LO SKETCH ARDUINOREMOTE

Ora che tutti i comandi sono stati mappati passiamo a quello che sarà lo sketch definitivo, ossia *ArduinoRemote*, che non riportiamo in queste pagine perché troppo voluminoso: lo troverete sul nostro sito www.elettronica.in.it insieme agli altri file del progetto. Lo sketch è stato scritto ipotizzando di utilizzare la configurazione elettronica per l'invio dei segnali vista in **Fig. 5**.

Nel listato viene presa in esame la casistica più complessa, ossia l'utilizzo di array di grandezza diversa rappresentanti un segnale raw. Nelle prime righe possiamo settare, attraverso la variabile *delayAmongSignals*, il ritardo tra un l'invio di un comando ed un altro per adeguarlo il più possibile alle tempistiche di ricezione del nostro dispositivo. Per impostazione predefinita, il valore è 500 millisecondi, quindi, ad esempio, per alzare il volume di 5 unità, verranno impiegati 2,5 secondi. Successivamente abbiamo l'elenco di tutti i comandi che abbiamo annotato precedentemente. In questo caso è stato mappato un telecomando di un televisore con i relativi comandi principali come accensione/spegnimento, numero del canale, volume e tasto "OK". Quest'ultimo comando torna utile per velocizzare il cambio canale nel caso in cui il televisore attenda qualche secondo prima di effettuarlo.

Come già accennato, bisognerà sostituire i comandi presenti con quelli mappati e, in caso di codifica non raw, sarà necessario modificare anche le funzioni adottate per l'invio del segnale.

Di seguito, abbiamo la parte di configurazione di Fishino UNO come web server.

Andiamo ad analizzare i parametri che servono per il suo funzionamento.

▪ #define MY_SSID "Inserire SSID"

Inserire il nome (SSID) della rete WiFi.

PROTOCOLLO	NUMERO DI BIT
NEC	32
RC5	12
RC6	20
SONY	12 o 20

Tabella 2
Numero di bit
dei protocolli
più in uso.



↓ Listato 2

```
//ArduinoRemoteSender by Carlo Palumbo

#include <IRremote.h>

//Inserire valore esadecimale altrimenti assegnare 0
unsigned int onHex = 0xE0E016E9;

//Inserire valore RAW altrimenti assegnare {}
unsigned int onRaw[68] = {4500, 4400, 600, 1650, 600, 1600, 600, 1600, 600, 550, 550, 550, 600, 500, 600, 500, 600, 500, 600, 1650,
600, 1600, 600, 1600, 600, 550, 550, 550, 600, 500, 600, 500, 600, 500, 600, 500, 600, 550, 600, 1600, 600, 500, 600, 500, 600, 550,
600, 500, 600, 500, 600, 1600, 600, 1650, 600, 500, 600, 1600, 600, 1650, 600, 1600, 600, 1600, 600, 1650, 600};

//Se vogliamo inviare il segnale RAW true altrimenti false
bool isRaw = false;

// Create IR Send Object
IRsend irsend;

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

void loop() {

  //Se legge la lettera o sulla seriale invia il segnale
  if (Serial.read() == 'o') {
    if (isRaw){
      // Inserire il nome del segnale, lunghezza dell'array e frequenza d'invio
      irsend.sendRaw(onRaw, 68, 38); // Accendi device
      Serial.println("Raw Inviato");
    }
    else {
      //Utilizzare sendRC5, sendRC6 o Sony se la vostra codifica è diversa
      //Inserire il nome del segnale e lunghezza di bit
      irsend.sendNEC(onHex, 32); // Accendi device
      Serial.println("Hex Inviato");
    }
  }
  delay(200);
}
```

- **#define MY_PASS "Inserire Password"**

Inserire la password della rete WiFi.

- **#define IPADDR 192, 168, 1, 114;**

#define GATEWAY 192, 168, 1, 1;

#define NETMASK 255, 255, 255, 0

Utilizzare un IP disponibile sulla propria rete e i giusti valori per il gateway e la netmask. Nel caso non siamo sicuri di tali valori, è possibile utilizzare il terminale e digitare *ipconfig* su Windows o *ifconfig* su Mac. In **Fig. 10** è mostrato un esempio di configurazione. Per quanto riguarda il gateway e la netmask ci basterà utilizzare gli stessi valori, per l'IP basterà cambiare l'ultimo gruppo di cifre a nostro piacimento.

- **FishinoServer server(3000)**

Viene definita la porta su cui rimarrà in ascolto

la scheda. Cerchiamo di scegliere un numero abbastanza elevato ed evitare la porta 80.

Passiamo ora ad analizzare la funzione *loop*.

Le prime righe lasciano la scheda in attesa di un client, il quale, una volta connesso, inoltra la sua richiesta che viene inserita in un buffer per poi essere analizzata con la funzione *processLine(buf)*. Quest'ultima si assicura che sia una richiesta di tipo GET e successivamente copia tale richiesta nella variabile *command*. Seguono i controlli per verificare se la richiesta corrisponde ad un'azione da eseguire. Il primo è immediato, infatti controlla che i primi sei caratteri in *command* corrispondano alla stringa *"turnon"* per poi inviare il segnale di accensione/spegnimento al televisore. Il secondo



Scheda LAN wireless Wi-Fi:

```
Suffisso DNS specifico per connessione: home
Indirizzo IPv6 locale rispetto al collegamento . : fe80::bc5c:2f74:95fe:3d31%4
Indirizzo IPv4. . . . . : 192.168.1.100
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.1.254
```

Fig. 10
Configurazione
di rete.

e il terzo controllo vengono utilizzati per riconoscere il comando di incremento e decremento del volume. Il loro funzionamento è speculare, quindi basterà analizzare il primo per comprendere il funzionamento anche del secondo. La prima istruzione che incontriamo fa avanzare il puntatore che rappresenta l'inizio della stringa del buffer di 17 caratteri. Questo numero è molto importante nel caso volessimo personalizzare i comandi da individuare: infatti, prendendo come esempio il nostro sketch e un'ipotetica richiesta per alzare il volume, avremo in *buf* una stringa di questo tipo: *GET /?volumeplus=2 HTTP/1.1*. Notate come i caratteri che precedono il numero 2, che rappresenta il nostro valore d'incremento, siano proprio 17. Essi, in seguito, vengono copiati all'interno di una variabile temporanea di dimensione 3, in questo modo potremo regolare il volume da un minimo di 1 ad un massimo di 99 unità (l'ultimo elemento dell'array è riservato al carattere di terminazione). Il passo successivo è trasformare la stringa in un intero e, attraverso un ciclo *for*, eseguire il comando d'invio del segnale al televisore tante volte quant'è il valore numerico della variabile *volume-Plus*. Infine, l'ultimo controllo serve per gestire il cambio del canale, il procedimento per ottenere le cifre è identico a quello del visto per la regolazione del volume ma c'è bisogno di qualche piccolo accorgimento in più.

La differenza, infatti, è che questa volta dovremo inviare le singole cifre e per farlo viene usata la funzione *changeChannel(int numChannel)*. Quest'ultima riceve in ingresso un intero che rappresenta il numero del canale che desideriamo e, mediante un algoritmo, salva le singole cifre in un array inserendo uno 0 nel caso in cui il numero contenga meno cifre rispetto al massimo fissato (5). Quindi, se richiedessimo il canale 208, il nostro array conterrebbe le seguenti cifre: 00208.

Non ci resta, allora, che inviare le singole cifre con la funzione *sendDigits(arrayChannel[i], numStart)* evitando gli 0 iniziali. Per farlo viene utilizzata una variabile booleana *numStart*, che diventa *true* soltanto dopo aver inviato la prima cifra diversa

da 0 per poi tornare a *false* alla fine del ciclo *for*. Pertanto, attraverso uno switch, vengono inviati i singoli segnali rappresentanti le cifre e infine anche il segnale che rappresenta "OK" sul telecomando. Al termine dell'invio si torna alla funzione di loop ed è possibile terminare la connessione rimanendo in ascolto per una successiva richiesta.

ABILITARE UNA PORTA DEL ROUTER

Per consentire alla piattaforma IFTTT di comunicare con Fishino UNO, dobbiamo fare in modo di reindirizzare tutto il traffico in entrata su una specifica porta del nostro router all'indirizzo IP della scheda. Purtroppo le procedure cambiano a seconda del modello di router, ma in generale basterà entrare nel pannello del proprio dispositivo digitandone l'indirizzo IP; di solito 192.168.1.1 per router utilizzati da TIM o Fastweb e 192.168.178.1 per router FRITZ!Box. Una volta entrati nel pannello di configurazione basterà trovare l'opzione "Abilitazioni", "Port Forwarding" o "Port Mapping" ed associare un nuovo dispositivo. Tutto ciò che bisognerà fare sarà aggiungere l'indirizzo IP o, se già presente in elenco, il nome della scheda Fishino UNO. Successivamente bisognerà selezionare il tipo di protocollo TCP e il numero di porta che abbiamo scelto di abilitare nello sketch, che nel nostro caso è 3000. Se viene richiesto il tipo di applicazione possiamo selezionare "Altro" o "Creare Virtual Server". Infine, possiamo salvare il tutto e utilizzare il sito

Fig. 11
Test partendo
dall'IP pubblico.





Fig. 12
Scelta
dell'account.

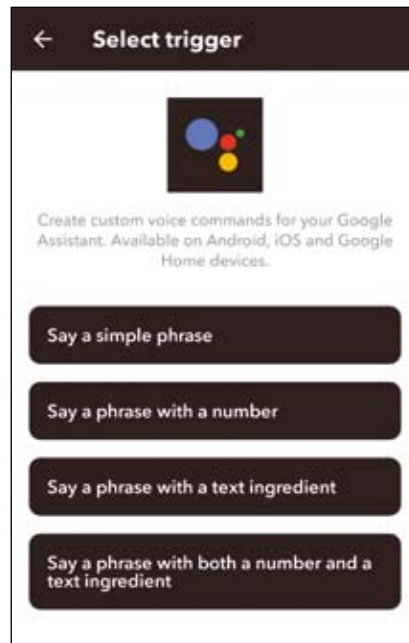


Fig. 13
Lista dei trigger
disponibili.



Fig. 14
Impostazione
del trigger.

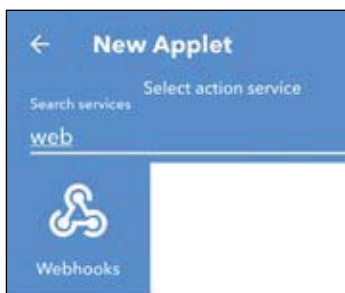


Fig. 15
Servizio
Webhooks.

www.ilmioip.it per individuare il nostro IP pubblico. Per effettuare un test, possiamo scrivere nella barra degli indirizzi del browser un indirizzo del tipo www.inserireippubblicotrovato/?channel=2. Cliccando su invio e puntando il LED nella direzione del nostro televisore, il canale verrà impostato su 2. Se la pagina web non risponde, possiamo utilizzare il sito www.yougetsignal.com/tools/open-ports/ e inserire il nostro indirizzo IP pubblico e la porta che abbiamo aperto, così da poter verificare se tutto è stato configurato correttamente (**Fig. 11**).

LA PIATTAFORMA IFTTT

Con la diffusione dei prodotti domotici è cresciuta tra i maker la necessità di integrare i propri dispositivi all'interno di servizi esistenti. Grazie a piattaforme come IFTTT (IF-THIS-THEN-THAT), però, è molto facile integrare, ad esempio, una lampada smart nell'ecosistema di Google Assistant, personalizzando i comandi. Essa permette la creazione di Applets che si innescano solo al verificarsi di una condizione prestabilita.

Le opzioni vanno dall'aggiungere sul calendario automaticamente un appuntamento al ricevere

una notifica quando un astronauta entra nello spazio! Nel nostro caso utilizzeremo il servizio Google Assistant per creare un trigger che si aziona con una frase personalizzata e il servizio Webhooks per inviare una richiesta http con metodo GET alla scheda che stiamo utilizzando come web server, in questo caso la Fishino UNO. Iniziamo quindi a iscriverci alla piattaforma, possiamo farlo sia online attraverso il sito www.ifttt.com sia scaricando l'app dal *Play Store* o *App Store*.

In questa sede utilizzeremo uno smartphone Android ma la procedura sarà molto simile anche se si sceglie di utilizzare il sito web. Quindi, dopo aver scaricato l'app, la prima cosa che ci verrà richiesta sarà di creare un account (**Fig. 12**), quindi scegliamo l'opzione che preferiamo e andiamo subito a creare la nostra prima applet. Per farlo basta andare nella tab in basso a destra *My Applets* e cliccare sul "+" in alto a destra e infine su "this". Si aprirà un pannello con la lista di tutti i servizi disponibili, scriviamo "*Google Assistant*" nella barra di ricerca e clicchiamo sull'icona corrispondente. Qui abbiamo la possibilità di scegliere il tipo di trigger che vogliamo utilizzare (**Fig. 13**); selezioniamo "*Say a simple phrase*".

Prima di poterlo fare, però, ci verrà richiesto di

← **Complete action**

URL
95.151.171.254:3000/?turnos

Surround any text with "<<>>" to escape the content

Method
GET

The method of the request e.g. GET, POST, DELETE

Content Type
text/plain

Optional

Body

Surround any text with "<<>>" to escape the content

Create action

Fig. 16
Impostazioni URL e richiesta.

connettere il nostro account associato a Google Assistant.

Una volta cliccato su *Connect* ed effettuato l'accesso, impostiamo il trigger come in **Fig. 14**. Inoltre, se non vogliamo alcuna risposta dall'assistente, basterà inserire un carattere speciale, come ad esempio delle virgolette, altrimenti possiamo scegliere una frase di risposta a nostro piacimento. È molto importante ricordare di non utilizzare frasi di trigger predefinite da Google altrimenti l'azione non verrà eseguita. Infine, selezioniamo *"Italian"* come lingua e clicchiamo su *"Create Trigger"*. Ora premendo su *"that"*, cerchiamo il servizio *"Webhooks"* (**Fig. 15**) e connettiamolo al nostro account premendo semplicemente *"Make a web request"* e in seguito su *"Connect"*.

Qui potremo impostare l'URL cui verrà inviata la richiesta GET che corrisponde al nostro indirizzo IP pubblico, seguito dalla formattazione della richiesta, che in questo caso serve ad accendere/spegnere il televisore e *"text/plain"* come *"Content Type"*. Si avrà dunque una situazione simile a quella in **Fig. 16**. Clicchiamo su *"Create Action"*, abilitiamo le notifiche se preferiamo e premiamo su *"Finish"*. La nostra prima Applet è pronta per essere utilizzata! Infine, creiamo altre tre Applet per il controllo

Fig. 17
Configurazione applet per incremento e decremento del volume.

← **Configure**

What do you want to say?
Imposta volume televisore + \$

Enter a \$ where you'll say the text ingredient

What's another way to say it? (optional)

Enter a \$ where you'll say the text ingredient

And another way? (optional)

Enter a \$ where you'll say the text ingredient

What do you want the Assistant to say in response?
Va bene

You can enter a \$ where you want to hear the text ingredient in the response

Language
Italian

← **Configure**

What do you want to say?
Imposta volume televisore meno \$

Enter a \$ where you'll say the text ingredient

What's another way to say it? (optional)

Enter a \$ where you'll say the text ingredient

And another way? (optional)

Enter a \$ where you'll say the text ingredient

What do you want the Assistant to say in response?
Va bene

You can enter a \$ where you want to hear the text ingredient in the response

Language
Italian

← **Complete action**

URL
254:3000/?volumeplus=

Surround any text with "<<>>" to escape the content

Method
GET

The method of the request e.g. GET, POST, DELETE

Content Type
text/plain

Optional

Body

Surround any text with "<<>>" to escape the content

Create action

← **Complete action**

URL
54:3000/?volumeminus=

Surround any text with "<<>>" to escape the content

Method
GET

The method of the request e.g. GET, POST, DELETE

Content Type
text/plain

Optional

Body

Surround any text with "<<>>" to escape the content

Create action

Fig. 18
Impostazione incremento e decremento volume con webhooks.

del volume e il cambio canale. Diversamente da prima, una volta premuto su *"this"* e cercato *"Google Assistant"*, clicchiamo su *"Say a phrase with a text ingredient"* che ci permetterà di impostare una nostra frase con all'interno quello che viene

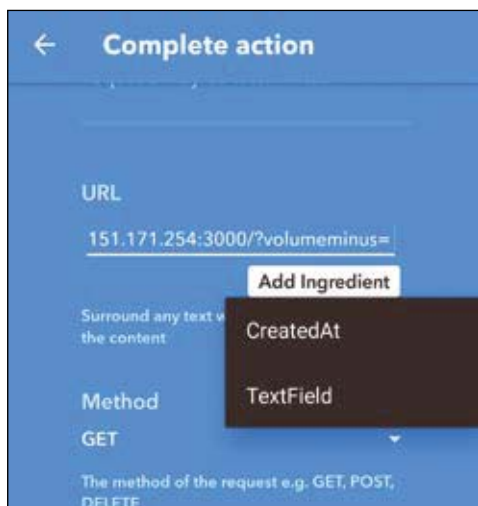


Fig. 19
Aggiunta
ingrediente.



Fig. 20
Applet per
il cambio
del canale.



Fig. 21
Applet per il cambio del canale.

chiamato "ingrediente", il quale rappresenta una variabile (ad esempio il numero di un canale televisivo) e viene rappresentato dal simbolo \$. Per quanto riguarda il servizio Google Assistant, configuriamo l'applet per l'incremento del volume, come mostrato a sinistra nella **Fig. 17** e quella per il decremento, come si vede a destra nella stessa figura.

Quindi riutilizziamo "Webhooks" come mostrato nella **Fig. 18**, rispettivamente a sinistra e a destra. Per poter aggiungere l'ingrediente: dopo il carattere "=" cliccare su "Add ingredient" e poi su "TextField" come in **Fig. 19**.

Per il cambio del canale basterà creare l'ultima Applet e configurarla come in **Fig. 20** e in **Fig. 21**. È importante notare come il modem potrebbe cambiare indirizzo IP pubblico, costringendoci a configurare nuovamente il servizio "Webhooks" con il nuovo indirizzo IP anche se questa eventualità si manifesta raramente.

CONCLUSIONI

Con questo progetto si è visto come comandare un televisore ma non ci sono limiti alla fantasia. Possiamo infatti adattarlo per controllare un lettore DVD, un condizionatore, un qualsiasi dispositivo IR, anche tutti insieme con un'unica scheda. È anche possibile aggiungere al codice intere

azioni, come ad esempio entrare nel menu del televisore e impostare lo spegnimento automatico dopo un'ora. Infine, è utile ricordare che per questo progetto è stata utilizzata una Fishino UNO ma nulla ci vieta di utilizzare qualsiasi altra scheda con a bordo un ESP8266 come ad esempio la nodeMCU, adattando ovviamente lo sketch. Inoltre, anche per quanto riguarda il server, possiamo utilizzare un Raspberry PI con servizi come *ngrok* che aggiunge complessità e costi al progetto ma ci permette di avere un indirizzo fisso senza il rischio che l'IP pubblico possa cambiare. □

Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board Fishino Uno (cod. FISHINO UNO) è in vendita a Euro 36,00, il led trasmettitore IR (cod. LED51R) è disponibile a Euro 0,60, mentre il led ricevitore IR (cod. IR38DM) costa Euro 2,80. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Conosci la posizione in tempo reale!

Localizzatori GPS, GSM/GPRS per tutte le esigenze in grado di monitorare in tempo reale l'auto, la moto, la bicicletta, i bagagli e tanto altro.



cod. TK102

€59,00

LOCALIZZATORE TASCABILE GPS/GSM-GPRS

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

LOCALIZZATORE GPS, GSM/GPRS PER VEICOLI

Localizzatore veicolare dotato di radiocomando per il controllo delle funzioni, e microfono per l'ascolto ambientale. È in grado di trasmettere le proprie coordinate (latitudine e longitudine) via SMS verso telefoni cellulari, oppure tramite tecnologia GPRS, ad un server web dedicato.

cod. FR617

€69,00



Prezzi IVA inclusa.

LOCALIZZATORE GPS, GSM/GPRS PER BICICLETTA

GPS, GSM/GPRS per bicicletta. Si inserisce nel tubo dello sterzo o sotto il sellino e rimane così completamente occultato. Può inviare le proprie coordinate (latitudine e longitudine) via SMS verso telefoni cellulari, oppure, tramite tecnologia GPRS, ad un computer opportunamente configurato.

cod. FR618

€89,00



trackimo

LOCALIZZATORI SATELLITARI GPS/GSM CON FUNZIONE SOS, CONTROLLABILI DA SMARTPHONE E PC



cod. TRACKIMO

€119,00

Scopri i localizzatori della famiglia TRACKIMO, piccoli, versatili e facilmente occultabili. Potrai tenere sotto controllo la posizione di persone, animali o cose, attraverso smartphone, tablet (iOS e Android) o PC. Dispongono 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.

cod. TRACKIMOMINI

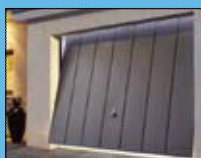
€149,00



CON TUTTI I PRODOTTI TRACKIMO HAI LA SIM CON 12 MESI DI SERVIZIO INCLUSI!

Rinnovo annuo € 49,00 oppure mensile € 4,90.

Controlla i tuoi dispositivi da remoto tramite LAN e Internet.



NOVITÀ

€ 88,00

Cod. FT1333M

SCHEDA ETHERNET CON 4 USCITE A RELÉ, 8 I/O DIGITALI E 4 INGRESSI ANALOGICI

- * 4 uscite a relé N.O. e N.C. con LED di stato;
- * Invio allarmi via e-mail;
- * Gestione uscite relé in modalità manuale o tramite programmazione oraria;
- * Possibilità di associare un ingresso digitale all'uscita relé di una scheda remota;
- * Alimentazione 7-18 Vdc;
- * Dimensioni 107 x 100 x 20mm;
- * Web server integrato;
- * Fornita già montata e collaudata.

SCHEDA ETHERNET CON 8 RELÉ 16A

- * 8 uscite a relé N.O. e N.C. con LED di stato;
- * Alimentazione 12Vdc;
- * Dimensioni 143 x 84 x 20mm;
- * Fornita già montata e collaudata.

€ 96,00

Cod. ETH008

€ 68,00

Cod. ETH-RLY02

SCHEDA ETHERNET CON 2 RELÉ 16A

- * 2 uscite a relé N.O. e N.C. con LED di stato;
- * Alimentazione 12Vdc;
- * Dimensioni 85 x 66 x 20mm;
- * Disponibile applicazione gratuita per Android;
- * Fornita già montata e collaudata.

SCHEDA ETHERNET CON 4 RELÉ 16A

- * 8 I/O digitali e 4 ingressi analogici;
- * 4 uscite a relé N.O. e N.C. con LED di stato;
- * Alimentazione 12Vdc;
- * Dimensioni 143 x 84 x 20mm;
- * Disponibile applicazione gratuita per Android;
- * Fornita già montata e collaudata.

€ 96,00

Cod. ETH484



Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

www.futurashop.it

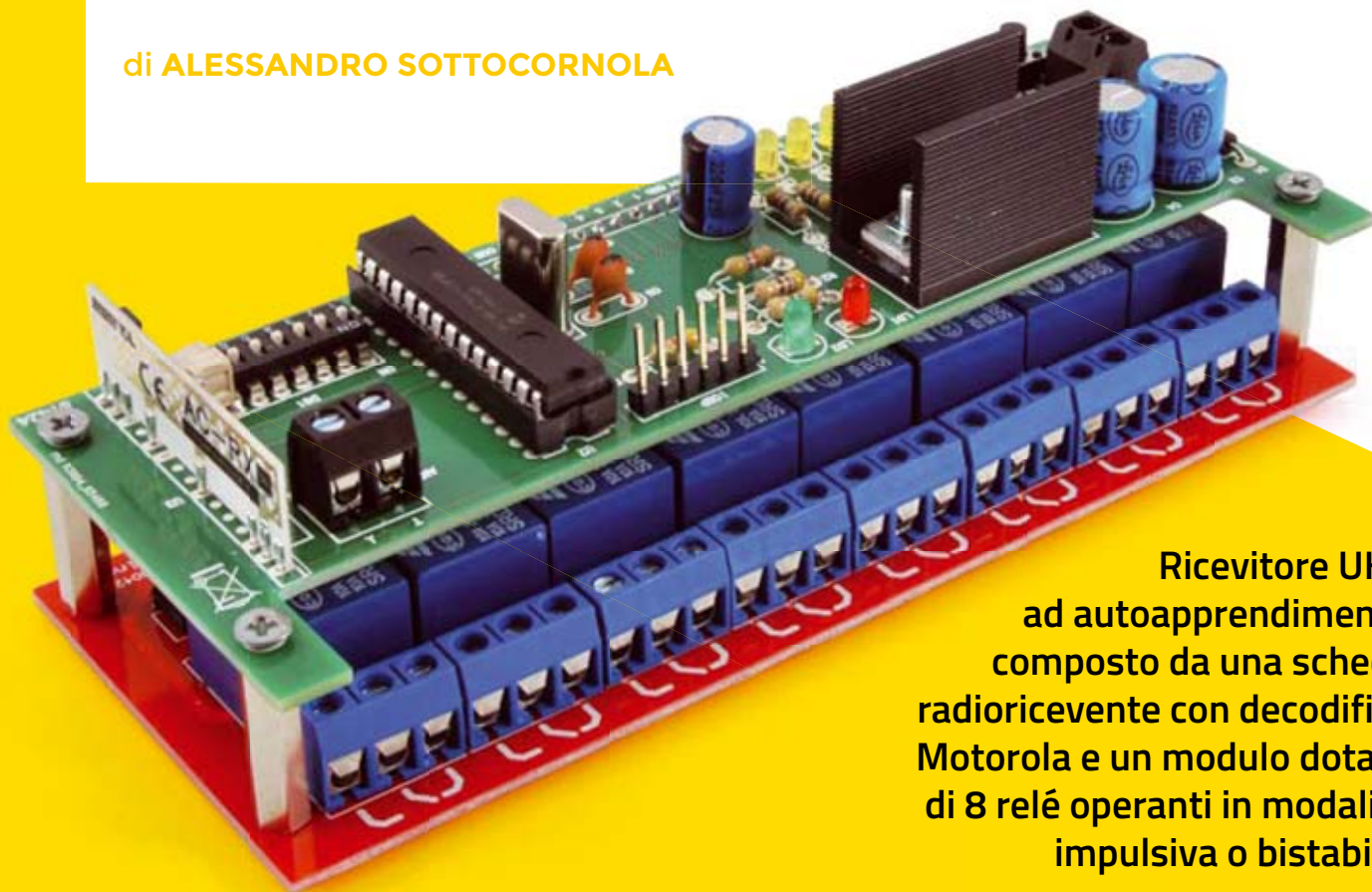
Futura Group srl
Via Adige, 11 • 21013 Gallarate (VA)
Tel. 0331/799775

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

RADIOCOMANDO

8 CH A 433MHZ

di ALESSANDRO SOTTOCORNOLA

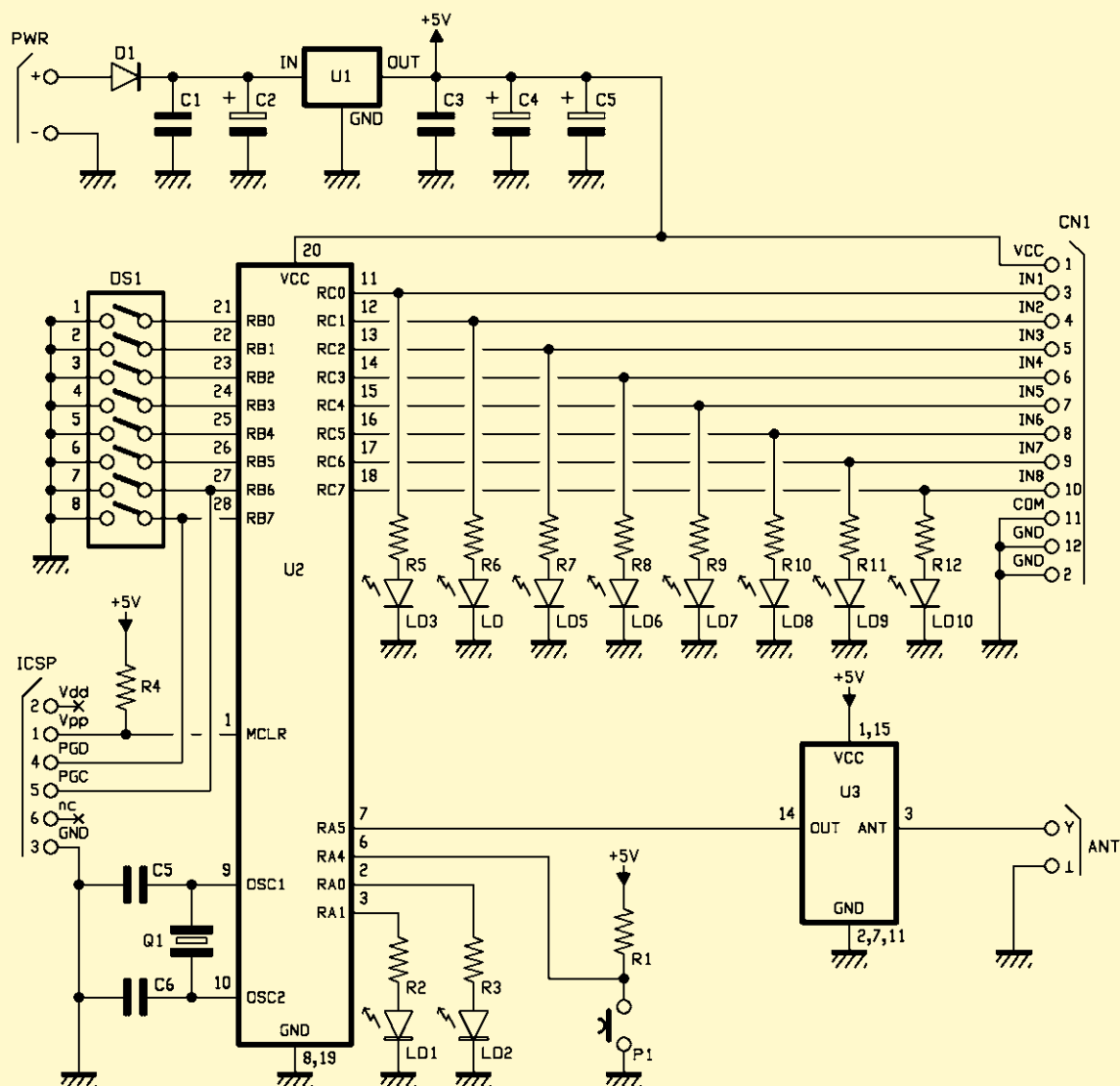


**Ricevitore UHF
ad autoapprendimento
composto da una scheda
radioricevente con decodifica
Motorola e un modulo dotato
di 8 relé operanti in modalità
impulsiva o bistabile.**

In molte delle applicazioni domotiche e in generale dove occorre esercitare un controllo a distanza via radio di molteplici apparati installati nello stesso luogo, i classici trasmettitori palmari mono e bicanale non sono particolarmente pratici, perché utilizzando i comuni radiocomandi (TX/RX) bisognerebbe installare più d'un ricevitore, con un evidente incremento dei costi, dell'ingombro e della complessità, giacché ciascuna unità ricevente ha parti ridondanti che potrebbero essere messe in comune: ad esempio il ricevitore RF. Ecco perché sarebbe utile disporre di un sistema radiocomando unico e, in particolare, di un'unità ricevente che supporti tutti i canali occorrenti al sistema; un esempio è il ricevitore descritto in queste pagine, che consente la gestione a distanza, tramite link radio AM a 433,92 MHz, di

8 utilizzatori gestibili con il contatto pulito di un relé ciascuno. Si tratta quindi di un sistema di radiocomando nel quale possiamo abbinare a una ricevente con otto uscite, un'unica trasmittente a otto canali, due da quattro, quattro bicanale oppure otto trasmettitori monocanale, ovvero combinazioni di essi, perché il bello del ricevitore è che apprende il codice a prescindere da quale sia la trasmittente da cui proviene. Grazie a un'opportuna interfaccia a relé, il ricevitore può commutare utilizzatori sia funzionanti in bassa tensione continua, sia alla tensione di rete 230Vca, che assorbano fino a 10 ampere ciascuno.

La codifica utilizzata è la standard Motorola MC14502x, quindi consente un massimo di 19.683 combinazioni per ciascun canale, a meno che non utilizzate dei TX pluricanale; infatti in questo caso le combinazioni vanno divise per il numero di



canali complessivo, ma tale limite non deriva dal ricevitore (che, come vedremo, tratta individualmente ogni singolo codice, a prescindere dal tipo di trasmettitore che lo genera) quanto piuttosto dall'encoder del trasmettitore. Per come funziona, il nostro ricevitore, rispetto a un dispositivo tradizionale, permette di sfruttare meglio la codifica: infatti, se a bordo avessimo un singolo decoder, come ad esempio l'MC145028, decodificare 8 canali significherebbe destinare gli ultimi due bit three-state alla definizione dei canali e lasciare i 7 restanti alla codifica vera e propria della trasmissione; invece nel ricevitore che vi proponiamo la decodifica viene gestita da un microcontrollore e ciascun canale viene trattato indipendentemente dagli altri, vale a dire che il

codice che il relativo TX trasmette è memorizzato distintamente.

Ecco quindi che potremmo abbinare otto trasmettenti monocanale a codifica Motorola MC14502x, ognuna delle quali può essere impostata scegliendo tra 19.683 combinazioni! Questo è il principale vantaggio del nostro sistema, che permette di mantenere un'elevata sicurezza contro l'effrazione pur impiegando una codifica che, se vogliamo, può essere considerata datata e poco efficace di fronte alle più avanzate rolling-code.

L'apprendimento si effettua singolarmente per ciascun canale, il che apre la strada a molte possibilità di utilizzo, in quanto possiamo abbinare a ciascun relé d'uscita un TX monocanale o un canale di un bicanale. Il firmware del microcontrollore che go-

verna il ricevitore fa in modo che un telecomando non possa comandare più relè. Ma lasciamo da parte questo ragionamento e andiamo subito a vedere di cosa si tratta, analizzando il sistema, che è composto da un'unità ricevente descritta in queste pagine e da una trasmittente palmare che si compera già fatta e che perciò esula da questa trattazione.

L'unità ricevente è a sua volta composta da un circuito stampato da autocostruire e del quale nei prossimi paragrafi descriviamo lo schema elettrico, oltre che da una scheda attuatrice di potenza a relé, che è un prodotto finito acquistabile dalla Futura Elettronica (on-line su www.futurashop.it) con il codice 2846-RELAY8CH. Il circuito da autocostruire è il blocco radoricevente e di decodifica Motorola, nel quale abbiamo integrato anche lo stadio alimentatore per tutto l'insieme.

SCHEMA ELETTRICO

Concentriamoci ora sul circuito, meglio descritto dallo schema elettrico che trovate in queste pagine, che si basa su un modulo ricevente ibrido come sezione RF e su un microcontrollore Microchip PIC16F876A-I/SP in funzione di decoder a standard Motorola MC14502x e gestore dello stato di funzionamento delle uscite. Il tutto è completato da un regolatore di tensione che consente di ricavare i 5 volt per far funzionare la logica on board e la scheda relé esterna.

Il modulo radoricevente è un circuito integrato ibrido AC-RX2/CS prodotto dalla Aurel, dotato di front-end con impedenza d'antenna di 50 ohm, amplificatore d'ingresso e stadio di sintonia super-rigenerativo a basse emissioni spurie (-65 dB), contraddistinto da elevata sensibilità (-100 dBm) e discreta selettività (la banda passante a -3dB è di 2 MHz) assicurata da un filtro ceramico; all'interno di questo ibrido, a valle dello stadio di sintonia si trova il demodulatore d'ampiezza OOK (On Off Keying) che fornisce al pin 13 il segnale demodulato così com'è e al 14 il segnale demodulato ma squadrato da un comparatore basato sull'integrato LM358, che consente di ricostruire gli impulsi trasmessi dal TX palmare con i fronti di salita e discesa ben squadrati, come richiesto dal microcontrollore del circuito per assicurare un'efficace decodifica.

Il micro U2 riceve al pin 7 (I/O RA5 configurato come input privo di pull-up) il segnale fornito dal piedino 14 dell'ibrido U3 e lo decodifica grazie a un'apposita routine, la quale viene avviata dal main program quando, girando in loop, verifica una transizione di livello nel segnale in arrivo, ovvero il

bit di start dell'invio dei dati.

Se il codice contenuto nella stringa ricevuta e demodulata dall'AC-RX2/CS non è tra quelli presenti nella memoria EEPROM dedicata, il firmware torna in ascolto sul piedino 7, mentre se figura tra quelli precedentemente memorizzati, il microcontrollore porta da 0 a 1 logico il livello dell'uscita corrispondente, tra quelle del registro RC, che il firmware dedica al controllo della scheda relé; per l'esattezza RC0 comanda il primo canale di tale scheda, RC1 il secondo e così via, fino ad RC7 che gestisce l'ottavo canale. I segnali di comando giungono alla scheda relé tramite il pin-strip CN1, ciascuna linea del quale fa capo a una linea IN corrispondente a un I/O del registro RC7. Sullo stesso connettore, che peraltro serve a innestare la nostra scheda su quella a 8 relé, viaggia l'alimentazione per la stessa scheda a relé, ossia Vcc (+5V) e GND, ossia la massa comune, cui si connette anche il COM (linea comune) degli ingressi di comando.

Oltre a pilotare il relativo ingresso della scheda relé, i pin del registro RC accendono ciascuno un LED di segnalazione, che si illumina in corrispondenza del livello logico alto e la cui corrente viene limitata da un resistore postogli in serie; grazie all'oculata scelta del valore della resistenza, la corrente erogata a ciascuno dei LED non abbassa particolarmente il livello della tensione di uscita del microcontrollore e ciò assicura sempre il corretto comando dei relé.

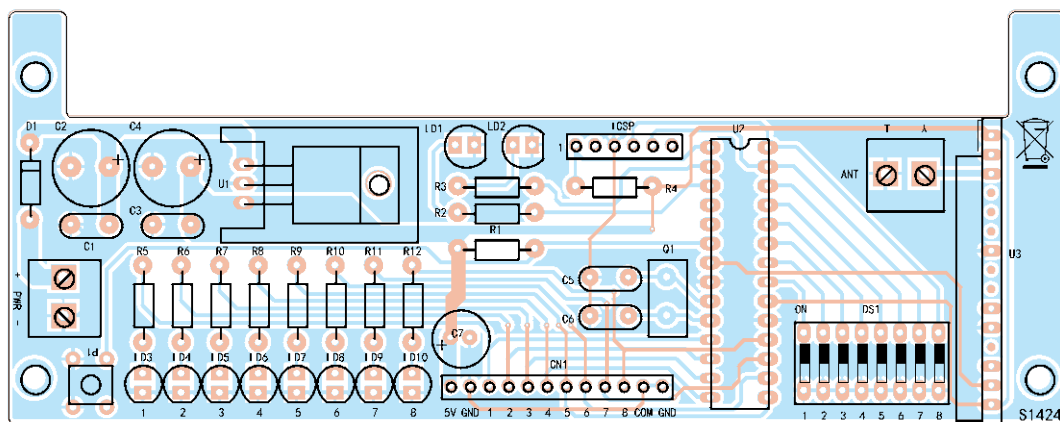
Va notato che se, come accennato a inizio articolo, abbiniamo due uscite a un singolo TX monocanale o allo stesso codice di un pluricanale, alla ricezione del comando il firmware va a comandare l'uscita che viene per prima in ordine, così da evitare la duplicazione del comando.

Rimanendo sul microcontrollore, spostiamoci sul port B, i cui otto bit vengono utilizzati per acquisire lo stato logico dei dip-switch che serviranno in apprendimento a selezionare il canale cui associare il codice trasmesso e in funzionamento ordinario a stabilire se il relativo relé dovrà operare in modalità bistabile o monostabile (l'uscita sarà attiva fino a quando il pulsante del trasmettitore non viene rilasciato e comunque per un tempo minimo di 500mS). Siccome il registro RB consente di attivare il pull-up interno per i suoi pin quando funzionano da input, lo abbiamo settato in modo da risparmiarci di dover connettere otto resistori esterni con pari funzione.

Procediamo l'analisi degli I/O con RA0 ed RA1, che il firmware inizializza come uscite per il comando dei LED di segnalazione da utilizzare, ad esempio,

CARATTERISTICHE TECNICHE

-  **Tensione di alimentazione:**
12 Vcc
-  **Corrente max. assorbita:**
250 mA
-  **Frequenza di lavoro:**
433,92 MHz
-  **Sensibilità RF:**
-100 dBm
-  **Codifica:**
19.683 combinazioni
-  **Canali/uscite:**
8
-  **Scambio relé:**
10 A/230 VCa
-  **Modalità uscite:**
monostabile e bistabile



Elenco Componenti:

R1, R4: 4,7 kohm
R2, R3, R5, R6, R7, R8, R9, R10, R11,
R12: 470 ohm
C1, C3: 100 nF ceramico
C2, C4: 470 µF 25 VL elettrolitico

C5, C6: 22 pF ceramico
C7: 220 µF 25 VL elettrolitico
D1: 1N4007
LD1: LED 3 mm rosso
LD2: LED 3 mm verde

LD3, LD4, LD5, LD6, LD7, LD8, LD9,
LD10: LED 3 mm giallo
P1: Microswitch
Q1: Quarzo 4 MHz
U1: 7805

durante l'apprendimento dei codici e il cui funzionamento verrà spiegato tra breve. Abbiamo poi RA5, inizializzata come ingresso senza pull-up, che nel caso specifico serve a leggere lo stato del pulsante P1 che avvia l'autoapprendimento; il pull-up è assicurato dalla resistenza R1.

Il funzionamento del microcontrollore è scandito dal clock generato dall'oscillatore interno, sincronizzato mediante il quarzo collegato tra i pin OSC1 e OSC2 e i rispettivi condensatori ceramici (C5 e C6) connessi a massa; in questo caso, essendo il quarzo da 4 MHz, la frequenza di clock è di pari valore.

Per la programmazione in-circuit, al connettore ICSP portiamo il piedino 1 (MCLR) e la linea d'alimentazione +5V (più la massa), oltre ai pin condivisi PGD (comune a RB7) e PCG (comune a RB6). Le 8 uscite possono funzionare sia in monostabile che bistabile, impostabili liberamente per ogni uscita.

Ad ogni singola uscita è abbinato un LED di stato da 3 mm (di colore giallo) il quale se sarà acceso indicherà l'uscita attivata (relè eccitato), mentre se risulterà spento indicherà che l'uscita corrispondente sarà a riposo.

Terminiamo il discorso sul microcontrollore precisando che il firmware, subito dopo l'inizializzazione degli I/O e prima ancora di porsi in ascolto sulla linea RA5 verifica lo stato dei dip-switch e quindi quello del pulsante P1: se quest'ultimo è premuto

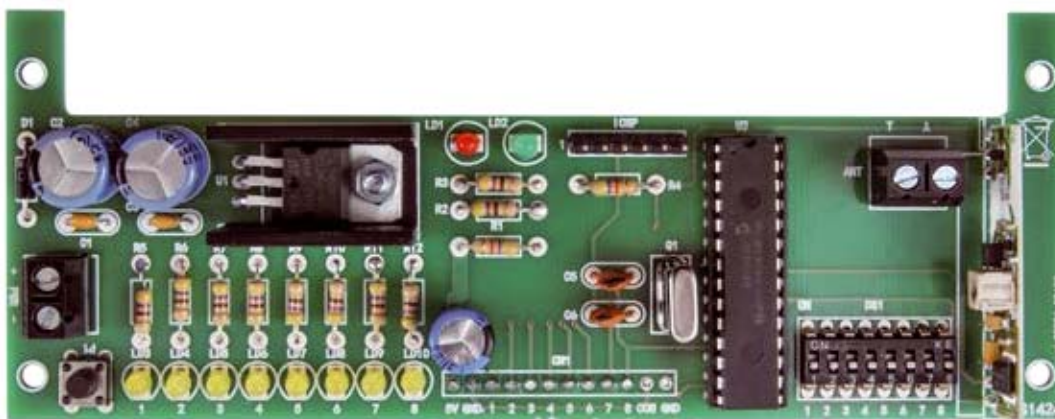
verifica che sia ad ON ed entra in apprendimento, altrimenti si porta nel normale funzionamento. Concludiamo l'analisi del ricevitore del radiocomando con il blocco di alimentazione, che riceve 12V (meglio se stabilizzati) attraverso la morsettieria PWR e la passa, tramite il diodo di protezione (dall'inversione di polarità) D1 e il filtro capacitivo formato da C1 e C2, al regolatore integrato a tre terminali U1 (è il canonico 7805), il quale ricava 5 volt perfettamente stabilizzati con cui funzionano il microcontrollore, il modulo ibrido radiorecettore e la scheda a relé collegata tramite il connettore CN1. La linea dei 5V è accuratamente filtrata dagli elettrolitici C4 e C5 e dal multistrato C3, opportunamente distribuiti sul PCB.

LA SCHEDA A RELÉ

Come accennato, i relè non sono presenti a bordo scheda, ma abbinando la scheda RELAY8CH le 8 uscite diventano a relè. Nel caso di abbinamento delle due schede, quella a relè viene installata sotto a quella di nostra realizzazione sfruttando il connettore pin strip 10 vie, mentre le altre 2 vie rimanenti rimarranno libere.

La scheda a relé è il modulo 8 relé 5 Vdc 10A codice 2846-RELAY8CH della Futura Elettronica. Si tratta di una scheda dotata di 8 relé che consente di attivare/disattivare il carico ad essi collegato, il quale può assorbire sino a 10 A e funzionare sia a 250 Vca che a 30 Vcc. Per ciascun relé vengono





U2: PIC16F876A-I/SP (MF1424)

U3: AC-RX2/CS

DS1: Dip-Switch 8 vie

ANT, PWR: Morsetto 2 vie passo 5.08 mm - Vite 10 mm 3 MA

CN1: Strip femmina 10 vie

ICSP: Strip maschio 6 vie

Varie:

- Vite 10 mm 3 MA

- Dado 3 MA

- Zoccolo 14+14

- Dissipatore TO220

- Distanziale metallico F/F 20 mm (4 pz.)

- Vite 8 mm 3 MA (8 pz.)

- Circuito stampato S1424 (140x55mm)

portati alla morsettiera d'uscita tutti i contatti, ossia C, NC ed NO (comune, normalmente chiusi e normalmente aperti). Gli ingressi della board sono tutti optoisolati (hanno in comune la linea COM, che colleghiamo alla massa della scheda del ricevitore) e per ciascuno è previsto a bordo un LED di stato; ogni relé si attiva quando l'input corrispondente è a livello logico alto, che nello specifico corrisponde a una tensione di valore compreso tra 1,5 e 5 V, pertanto la board viene pilotata con livelli logici come quelli forniti dagli I/O del nostro microcontrollore (l'assorbimento su ciascuna linea d'ingresso è dell'ordine di qualche milliampere, giusto quel che occorre ad accendere il LED interno

a ciascun optoisolatore) o da quelli di qualsiasi scheda Arduino.

REALIZZAZIONE PRATICA

Passiamo adesso alla costruzione del nostro ricevitore per radiocomando, che riguarda solo la scheda ricevitrice della quale nel nostro sito www.elettronica.in.it trovate, insieme agli altri file del progetto, le tracce lato rame a grandezza naturale da stampare su lucido o acetato per ottenere le pellicole necessarie a preparare il circuito stampato per fotoincisione.

Una volta inciso e forato il PCB, montatevi i pochi componenti richiesti a cominciare da resistenze e

LA LAMPADA SI COMANDA DALL'iPhone

La domotica passa sempre più dal controllo a distanza in wireless tramite gli smartphone e i loro canali di comunicazione locale: Bluetooth e WiFi. Proprio con il WiFi funziona la lampada smart Xiaomi Mijia Bedside Lamp 2, che conserva le funzionalità intrinseche delle lampade del produttore cinese, integrando nel design una barra touch posizionata verticalmente su un lato, che consente di regolare



la luminosità e la tonalità della luce, ottenuta da LED RGB capaci di un flusso luminoso di 400 lumen. La lampada può essere gestita dall'app companion di Yeelight e Mi Home (quindi di Xiaomi), però è compatibile con il sistema Apple HomeKit senza dover ricorrere all'installazione di software di terze parti.

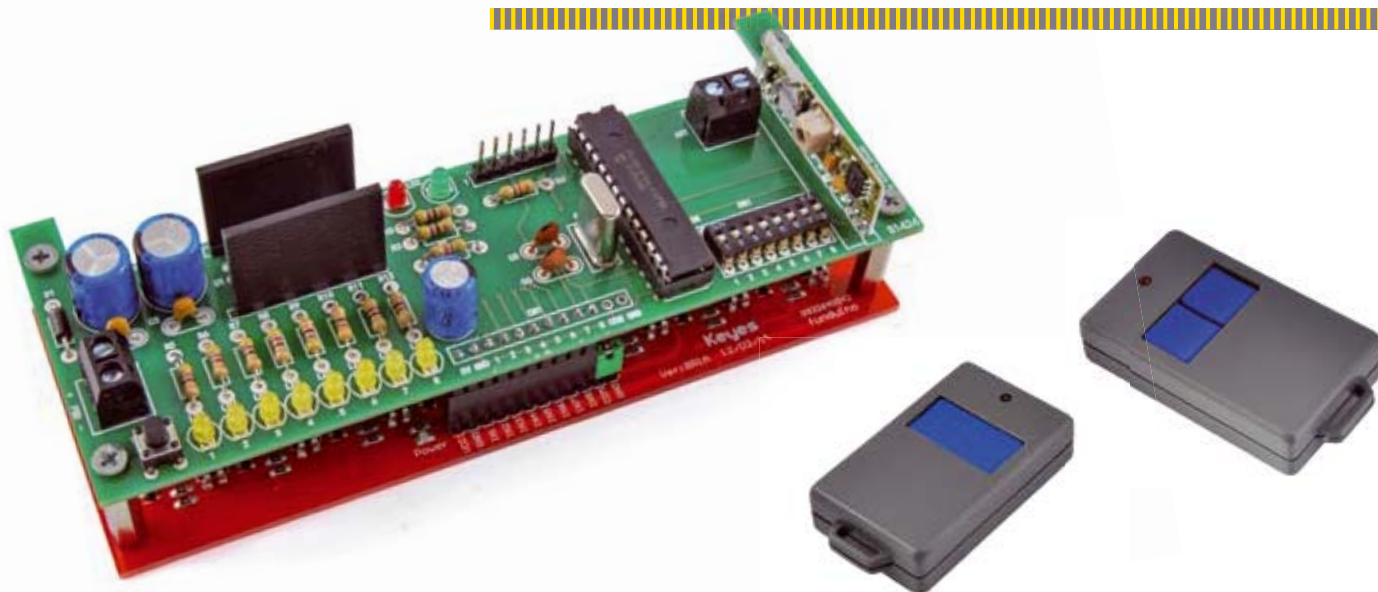


Fig. 1
Il nostro ricevitore
è adatto ai
trasmettitori
TX1CSAW e
TX2CSAW
codificati
Motorola.

diodi al silicio (D1) e LED (di cui va rispettata la polarità indicata nel piano di montaggio), procedendo con lo zoccolo per il microcontrollore, quindi con il dip-switch (da orientare con lo switch 1 rivolto allo zoccolo del PIC, fermo restando che nell'inserire lo zoccolo lo abbiate orientato correttamente), il pulsante e i condensatori (state attenti alla polarità degli elettrolitici); terminate con i transistor, il regolatore a tre terminali, il pin-strip CN1 (che va saldato dal lato componenti ma introdotto dal lato saldatore, giacché andrà introdotto nello strip femmina della scheda relé) e quello per l'ICSP, quindi le morsettiere.

Poi è la volta del modulo AC-RX che va inserito e saldato nelle rispettive piazzole, con il lato componenti rivolto all'interno dello stampato (non è comunque possibile montarlo al contrario, visto il layout del PCB e la disposizione dei rispettivi pin...) del quarzo per il microcontrollore (che non ha polarità) e del regolatore 7805 in contenitore TO-220, in cui piedini devono essere piegati ad angolo retto prima di introdurli nei rispettivi fori. Il regolatore va appoggiato sulla base e all'interno di un dissipatore di calore in alluminio sagomato ad "U" tipo il classico ML26-T0220, avente resistenza termica di 18 ± 20 °C/W, poi avvitato con un bullone 3MA e un dado che sia passante dall'aletta metallica del 7805 al lato saldature del PCB, in modo da bloccare il componente. Per agevolare lo smaltimento del calore sarà buona regola spalmare della pasta al silicone tra il 7805 e il dissipatore; fissato meccanicamente il gruppo U1-dissipatore, saldate i tre terminali alle rispettive piazzole.

Per tutte le fasi del montaggio aiutatevi con i piani di montaggio che trovate in queste pagine.

Il microcontrollore PIC16F876-I/SP, che può essere programmato sia mediante l'apposito program-

matore (e poi inserito nel rispettivo zoccolo stando attenti a orientarlo come indicato nel rispettivo disegno di montaggio) sia a bordo scheda tramite il connettore ICSP.

Una volta completata e programmata la scheda con ricevitore e decodifica dovete applicarla al modulo relé, al quale nel frattempo avrete avvitato una colonnina esagonale lunga 20 mm in ciascuno dei fori perimetrali di fissaggio, innestando il pin strip nello strip femmina della scheda relé; spingete le schede una contro l'altra e stringete la scheda ricevente-decodifica con delle viti 3MA da avvitare nelle colonnine esagonali applicate al PCB del modulo relé. Notate che il circuito stampato della sezione ricevente e decodifica è stato volutamente disegnato con lo "scavo" per rendere accessibili le viti delle morsettiere del modulo relé.

Avete così completato il vostro ricevitore a 8 canali, al quale ora mancheranno solo due cose: un'antenna, che potrà essere un semplice spezzone di filo in rame isolato (ma spelato nel punto di collegamento alla scheda) serrato nel morsetto della morsettieria ANT rivolto verso l'ibrido AC-RX2/CS, nonché un contenitore in plastica adatto ad ospitarlo, lavorato in modo da rendere accessibili le morsettiere del modulo relé e quella dell'alimentazione. Se optate per lo spezzone di filo, dovrete sceglierlo di lunghezza pari a 17 ± 18 cm, che alla frequenza di lavoro del sistema corrispondono a $1/4$ d'onda. Potete anche optare per un'antenna esterna, come ad esempio uno stilo o una ground-plane, sempre accordata sui 433,92 MHz (433 ± 434 MHz) da collegare mediante uno spezzone di cavo schermato coassiale tipo RG59 e comunque per UHF; in questo caso la maglia di schermo andrà collegata dal lato della scheda ricevente al morsetto di massa e il conduttore centrale al morsetto che sta vicino

MICROINTERRUTTORE	APPRENDIMENTO	NORMALE UTILIZZO
1	memorizza, se P1 è premuto, il codice dell'uscita 1	OFF = uscita 1 a impulso ON = uscita 1 bistabile
2	memorizza, se P1 è premuto, il codice dell'uscita 2	OFF = uscita 2 a impulso ON = uscita 2 bistabile
3	memorizza, se P1 è premuto, il codice dell'uscita 3	OFF = uscita 3 a impulso ON = uscita 3 bistabile
4	memorizza, se P1 è premuto, il codice dell'uscita 4	OFF = uscita 4 a impulso ON = uscita 4 bistabile
5	memorizza, se P1 è premuto, il codice dell'uscita 5	OFF = uscita 5 a impulso ON = uscita 5 bistabile
6	memorizza, se P1 è premuto, il codice dell'uscita 6	OFF = uscita 6 a impulso ON = uscita 6 bistabile
7	memorizza, se P1 è premuto, il codice dell'uscita 7	OFF = uscita 7 a impulso ON = uscita 7 bistabile
8	memorizza, se P1 è premuto, il codice dell'uscita 8	OFF = uscita 8 a impulso ON = uscita 8 bistabile

Tabella 1
Funzionamento
dei dip-switch.

al modulo ibrido AC-RX. Dal lato dell'antenna, nel caso del semplice stilo connettete quest'ultimo al conduttore centrale e isolate con della guaina termorestringente la maglia di schermo, dopo averla ripiegata all'indietro per evitarne il contatto con il conduttore centrale e lo stilo, mentre se avete una ground-plane connettete lo stilo come spiegato e la calza di schermo al piano di massa.

Per far funzionare l'insieme serve un alimentatore in grado di fornire una tensione continua, meglio se stabilizzata, di 12 V e una corrente di almeno 250 milliampere, che sono l'assorbimento massimo con tutti i relé eccitati e tutti i LED accesi. Una volta completato il montaggio, il circuito è pronto per l'uso, ma naturalmente dev'essere abbinato ai trasmettitori, i quali, lo ricordiamo, sono i classici funzionanti in AM a 433,92 MHz e codificati Motorola MC145026x: per esempio i modelli TX1CSAW e TX2CSAW venduti dalla Futura Elettronica (www.futurashop.it) con i codici rispettivamente 8110-TX1CSAW e 8110-TX2CSAW. Vediamo dunque come si effettua l'abbinamento (autoapprendimento) dei codici e pertanto dei trasmettitori da usare insieme alla ricevente.

MEMORIZZAZIONE DEL CODICE DEL TRASMETTITORE

Prima di entrare nel dettaglio della procedura ricordiamo che la scheda non fa distinzione tra unità trasmittente e pulsanti, nel senso che in autoap-

prendimento memorizza il codice ricevuto nella posizione di memoria corrispondente al canale sul quale si sta effettuando l'apprendimento; questo significa che se si sta abbinando un TX bicanale, ciascuno dei due pulsanti verrà appreso per conto proprio.

Ciò detto vediamo come effettuare l'abbinamento: a scheda alimentata, portare ad ON il solo micro-interruttore (quello corrispondente all'uscita da abbinare) del dip-switch DS1, quindi premere e tenere premuto il pulsante P1 sulla scheda fin quando il LED rosso si accenderà fisso; a questo punto dovete premere il pulsante sulla trasmittente, stando a una distanza di almeno mezzo metro, per non saturare la sezione radioricevente. Quando il LED rosso sulla scheda inizierà a lampeggiare, la memorizzazione sarà avvenuta correttamente, quindi si potrà rilasciare il pulsante sulla scheda e quello sul telecomando.

Ripetete questa procedura per tutte le uscite che desiderate utilizzare impostando il canale desiderato con DS1, ricordando che è possibile memorizzare un solo codice identificativo del pulsante per ciascuna uscita a relé e che non potrà essere abbinato lo stesso codice di pulsante o trasmettitore su più uscite.

CANCELLAZIONE CODICE USCITA

Togliere alimentazione alla scheda, quindi portare ad ON i microinterruttori del dip-switch corrispon-



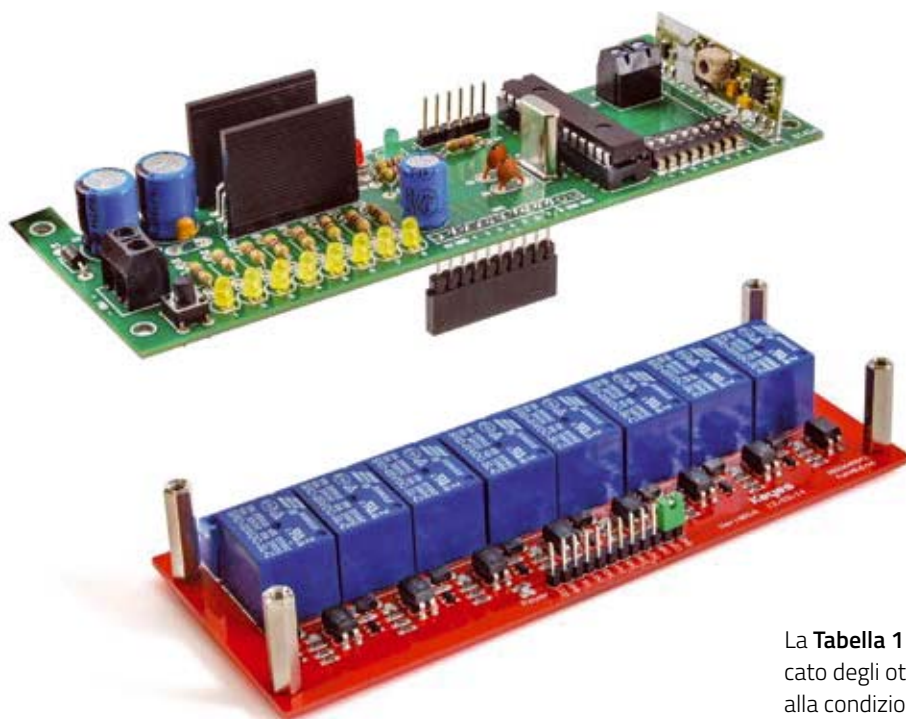


Fig. 2
Le due schede
componenti
il radiocomando
si innestano
mediante strip.

denti all'uscita dove si desidera disattivare il radiocomando, quindi premere e mantenere premuto il pulsante sulla scheda P1 e fornire alimentazione. Il LED rosso si accenderà fisso per circa 1 secondo ad indicare che la procedura di rimozione abbinamento è in corso.

Quando il LED rosso si spegnerà, l'unità ricevente sarà pronta per riprendere il normale funzionamento, vale a dire per ricevere i comandi e attivare i relé di conseguenza. Notate che si potranno portare ad ON anche più microinterruttori.

IMPOSTAZIONE MODALITÀ USCITA

Ogni micro interruttore del dip-switch DS1 corrisponde alla rispettiva uscita sulla scheda, quindi il dip 1 corrisponderà all'uscita 1, il dip 2 all'uscita 2 e così di seguito; questo vale non solo nell'autoapprendimento dei codici dei radiocomandi o loro pulsanti che comanderanno le uscite, ma anche per l'impostazione della modalità di attivazione dei relé corrispondenti.

Dunque, in autoapprendimento lo stato di ciascun dip-switch indica al microcontrollore che quando viene premuto P1 deve memorizzare il codice corrispondente al relé dell'OUT1 della scheda relé, mentre nel normale funzionamento portando ad ON il microinterruttore si imposterà il funzionamento bistabile, mentre sarà monostabile se lasciato ad OFF.

La **Tabella 1** chiarisce il funzionamento e il significato degli otto micro-interruttori del DS1 in base alla condizione di funzionamento.

La modalità dei relé di uscita può essere cambiata durante il funzionamento senza dover rimuovere e riapplicare l'alimentazione, come invece avviene per l'autoapprendimento.

CONCLUSIONI

Il radiocomando descritto in queste pagine è un progetto modulare adatto quando occorre gestire vari utilizzatori con un'unica unità di controllo; presenta il vantaggio di poter essere abbinato anche a più trasmettenti con codici distinti, così da rendere, ad esempio, selettivo l'accesso a un passaggio con tornello o cancello motorizzato, ovvero il controllo di un certo utilizzatore. □



Cosa occorre?

I componenti utilizzati per realizzare questo progetto sono disponibili presso Futura Elettronica.

Il ricevitore 8 ch ad autoapprendimento (cod. FT1424K) è in vendita a Euro 69,00, la scheda con 8 relé (cod. RELAY8CH) è disponibile a Euro 19,90, il trasmettitore 1ch (cod. TX1CSAW) costa Euro 18,00, quello 2ch (cod. TX2CSAW) è disponibile a Euro 19,50, mentre il quello a 4ch (cod. TX-4C-5026) costa Euro 20,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

STAMPANTE 3D4040



oggi ancora più ricca di accessori!

STAMPANTE 3D 40x40x40 cm in KIT

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

€ 999,00

Cod. **3D4040**

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



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



Box in plexiglass per
stampante 3D4040

VASTA GAMMA DI ACCESSORI!

SCOPRILI TUTTI SU
www.futurashop.it

Controller per stampa
autonoma con display grafico

NOVITÀ!

Cod. FT1147K
€ 34,00

Relè allo stato
solido per piatto riscaldato

Foglio riscaldato
in kapton 40x40cm

Cod. BOX3D4040
€ 198,00

Cod. FT1357K
€ 15,00

Cod. 3D4040HPLATE
€ 65,00

MOTORFISH: LO STEPPER NON SBAGLIA UN PASSO



Controller con retroazione per motore passo-passo in grado di verificare dinamicamente il posizionamento dell'albero e prevenire eventuali perdite di passo.



di **MASSIMO DEL FEDELE**

A

chi non è mai capitato di lanciare una stampa 3D complessa e lunga, magari un lavoro di ore o decine di ore, assentarsi e scoprire che, a lavoro quasi terminato, i motori della stampante hanno perso qualche passo col risultato di avere l'oggetto rovinato stampato male con degli strati non allineati tra

loro? Purtroppo non è infrequente e la cosa sorprendente (o irritante...) è che succede quasi sempre verso la fine della stampa, causando notevole spreco di materiale e di tempo. Siccome la stragrande maggioranza delle stampanti 3D in commercio utilizza dei motori passo-passo controllati ad anello aperto, la cosa è inevitabile perché il controller non ha modo di accertare se l'ordine impartito viene correttamente eseguito dal motore, ovvero se ad ogni impulso inviato il rotore avanza effettivamente di un passo.

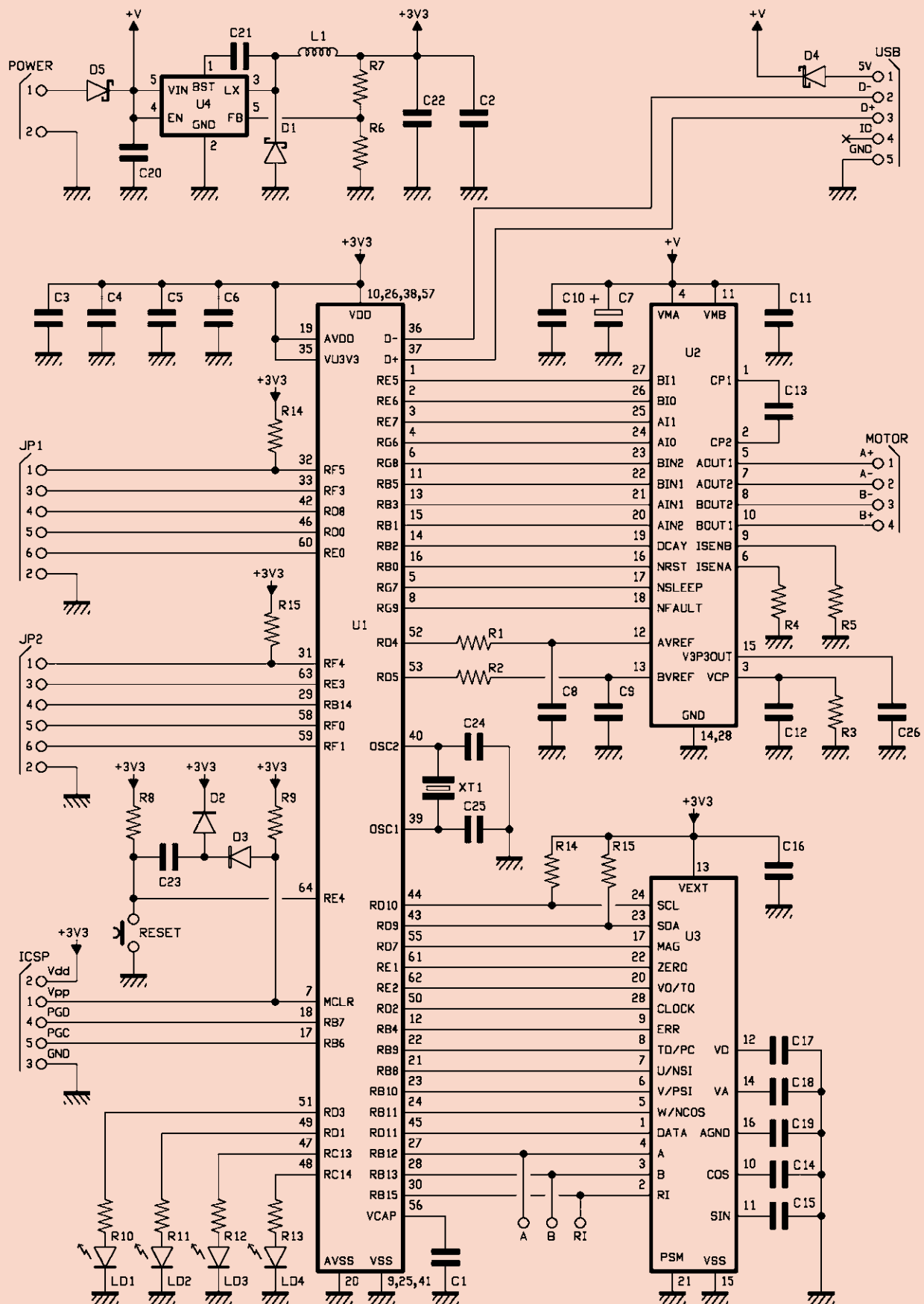
Per sopperire questa carenza abbiamo deciso di realizzare un controller, chiamato Motorfish, in grado di gestire localmente (a stretto contatto...) un motore passo-passo dotato di encoder, quindi di operare un controllo a reazione (o retroazione) in grado di verificare che la posizione angolare dell'albero dello stepper motor sia coerente con gli impulsi di pilotaggio. Prima di analizzare lo schema elettrico è opportuno chiarire come funzionano i due tipi di controllo di posizione dei motori, che sono ad anello aperto e ad anello chiuso.

Nel primo caso, anche noto come open loop, il controller invia un comando al motore senza conferma dello spostamento effettivo; se il motore per un qualsiasi motivo non è stato in grado di spostarsi (ad esempio se la testina di stampa ha incontrato un ostacolo oppure se la velocità di spostamento richiesta è troppo elevata) l'errore viene mantenuto e si accumula con eventuali altri errori successivi.

L'elettronica non ha alcun modo di conoscere la reale posizione del motore, col risultato di avere delle stampe magari "sfalsate" da metà lavoro in poi.

Nel controllo ad anello chiuso (closed loop, **Fig. 1**) viene impartito un comando di spostamento al motore, ma in questo caso un sensore di posizione sul medesimo è in grado di fornire una "conferma" dello spostamento avvenuto o un segnale di errore se questo è impossibile. Il controller può quindi decidere se fermare tutto, riprovare oppure andare avanti cercando di correggere il problema in seguito.

Il sistema ad anello chiuso ha parecchi vantaggi sul primo; innanzitutto, lo spostamento non dipende più dalla tipologia del motore, ma soltanto dalla risoluzione del sensore di



movimento (un **encoder**, abitualmente). Quindi non saremo più obbligati a lavorare per passi discreti (step) o micropassi (microstep) ma potremo avere un funzionamento continuo del motore, con minori vibrazioni meccaniche.

Un secondo vantaggio è che, non essendo a rischio di perdita di posizione, possiamo “spingere” il motore al massimo senza preoccuparci di nulla o quasi; se il motore non riesce a seguire i comandi l’elettronica se ne accorge ed è in grado di rimediare, sia riducendo velocità e/o accelerazione sia correggendo gli errori.

Il terzo vantaggio è un ridotto consumo di corrente. In un motore passo-passo a ciclo aperto siamo infatti obbligati a mantenere una corrente elevata negli avvolgimenti anche a motore fermo, per evitare movimenti indesiderati dovuti magari a masse in rapida decelerazione oppure a manomissioni involontarie del piatto della stampante; nel motore a ciclo chiuso, quando non è in movimento, possiamo ridurre a zero la corrente negli avvolgimenti, sicuri che in caso di spostamenti involontari l’elettronica sarà in grado di correggere immediatamente la posizione.

Ultimo, ma non in ordine di importanza, è la possibile semplificazione del motore stesso. Non sarà il nostro caso, visto che utilizzeremo comunque un motore passo-passo, ma volendo è possibile sostituirlo con un qualsiasi altro tipo di motore, anche a spazzole o brushless, cambiando semplicemente l’elettronica e/o il software di controllo. Come vedremo in seguito, il nostro **MotorFish** è in grado di controllare anche un motore a spazzole (**brushed**) modificando semplicemente il software del controller. I motori brushed utilizzano un solo “avvolgimento” (non è proprio così, ma questo è ciò che appare all’esterno), ed il nostro driver è in grado di pilotarne fino a 2; purtroppo non sarà possibile pilotare motori a 3 avvolgimenti.

SCHEMA ELETTRICO

Dopo il lungo preambolo sul controllo a retroazione passiamo a vedere come viene implementato, analizzando il circuito che lo realizza.

Iniziamo con la descrizione della sezione di alimentazione; per evitare di “portarci in giro” troppi fili abbiamo infatti previsto un piccolo alimentatore switching sulla scheda, in grado di fornire i 3,3 volt necessari all’elettronica digitale direttamente dalla tensione elevata utilizzata per pilotare il motore. I motori passo-passo, infatti, per compensare i ritardi dovuti all’elevata induttanza degli avvolgimenti, sono solitamente pilotati da driver in moda-

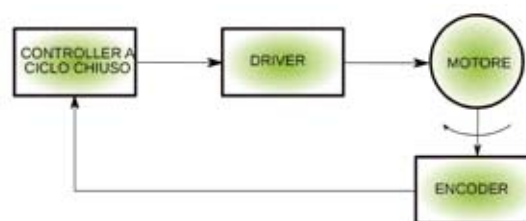


Fig. 1
Schematizzazione
del controllo ad
anello chiuso.

lità PWM con tensioni piuttosto elevate. Il nostro alimentatore è in grado di funzionare con tensioni in ingresso da circa 5-6 Volt fino a 42 volt massimi, coprendo quindi la gamma delle tensioni utilizzate abitualmente nei motori passo-passo.

Si tratta di un semplice convertitore buck (abbassatore), costituito da un integrato LMR14206, una piccola induttanza, un diodo e pochi altri componenti passivi.

Abbiamo scelto una circuizione switching poiché con una differenza elevata tra tensione in ingresso ed in uscita uno schema con regolatore lineare sarebbe molto inefficiente e richiederebbe un dissipatore di calore di elevate dimensioni, mentre nel nostro caso si raggiunge un’efficienza vicina al 90% con un riscaldamento praticamente nullo.

L’integrato lavora a 1,2 MHz, cosa che permette di utilizzare componenti di dimensioni ridottissime, soprattutto per quanto riguarda l’induttanza; per contro, è necessaria una cura meticolosa nello studio del circuito stampato, utilizzando collegamenti corti, ampie zone di massa e componenti molto vicini al circuito integrato.

La regolazione della tensione in uscita avviene attraverso il pin di feedback (FB), il quale è impostato internamente per una tensione di 0,765 volt; occorre quindi un partitore resistivo che fornisca quel valore quando la tensione in uscita raggiunge i 3,3 volt richiesti.

Il nostro partitore, costituito da R7 (4,4 kohm) e R6 (1,02 kohm) consente di ottenere una tensione in uscita pari a:

$$V_o = V_{FB} \cdot \frac{R_6 + R_7}{R_6} = 0.765 \text{ V} \cdot \frac{3.4 \text{ K} + 1.02 \text{ K}}{1.02 \text{ K}} = 3.315 \text{ V}$$

quindi molto vicina ai 3,3 V.

IL CONTROLLER

Per gestire tutti i componenti (che come vedremo a breve non sono pochi!) con la necessaria velocità di risposta abbiamo optato per un controller a 32 bit, per la precisione una PIC della serie 32MX, dotata di interfaccia USB per facilitarne la programmazione e la gestione tramite un PC.

Il bootloader presente sulla stessa ci permetterà di caricare il software in maniera molto semplice tramite l'IDE di Arduino o la nostra FishIDE, se preferite. La scrittura del software necessario è quindi molto semplice, esattamente come se si trattasse di una scheda Arduino-compatibile. Il clock necessario al funzionamento del controller è fornito da un quarzo a 20 MHz dal quale, internamente, vengono ricavate sia la frequenza di clock effettiva (80 MHz) che quella relativa al sottosistema USB (48 MHz).

IL DRIVER

Come driver per il motore abbiamo utilizzato un integrato della Texas Instruments, il DRV8843, parente stretto del più noto DRV8825, utilizzato in molte stampanti 3d. Questo, a differenza del DRV8825 non viene pilotato tramite gli abituali comandi di STEP, DIR e ENABLE, ma direttamente da impulsi PWM generati dal nostro controller (**Fig. 2**); ciò permette una notevole flessibilità di utilizzo, compresa la generazione di microstepping

molto fitti. Vediamo ora un po' più in dettaglio il driver: i due avvolgimenti del motore stepper sono denominati A e B; da qui i nomi dei relativi pin. Nella **Fig. 3** potete vedere lo schema interno (semplificato) di un canale del driver; la parte è relativa al canale A.

I pin importanti per la comprensione dello schema sono :

- **VMA** è l'ingresso di alimentazione del motore, che può andare da 8.2 a 45 Volt, da cui la massima tensione accettata dal nostro alimentatore switching, pari a 46 volt;
- **AIN1/AIN2** sono gli ingressi di controllo del ponte H che alimenta l'avvolgimento. Tramite questi pin è possibile controllare la tensione sulle uscite AOUT1 e AOUT2 secondo la **Tabella 1**.

In essa **Z** sta per 'uscita ad alta impedenza', ovvero l'equivalente della disconnessione dell'avvolgimento, **L** sta per tensione negativa ed **H** per tensione positiva.

L'integrato dispone all'interno di un controllo per evitare condizioni 'pericolose' nel ponte H, come ad esempio l'accensione simultanea dei due MOSFET superiori e di quelli inferiori, cosa che provocherebbe un rovinoso cortocircuito sull'alimentazione. Tramite questi 2 pin è quindi possibile fornire l'alimentazione all'avvolgimento nella direzione desiderata. Applicando un livello logico ai 2 pin l'avvolgimento viene alimentato dalla corrente

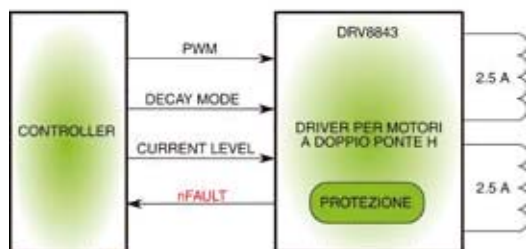
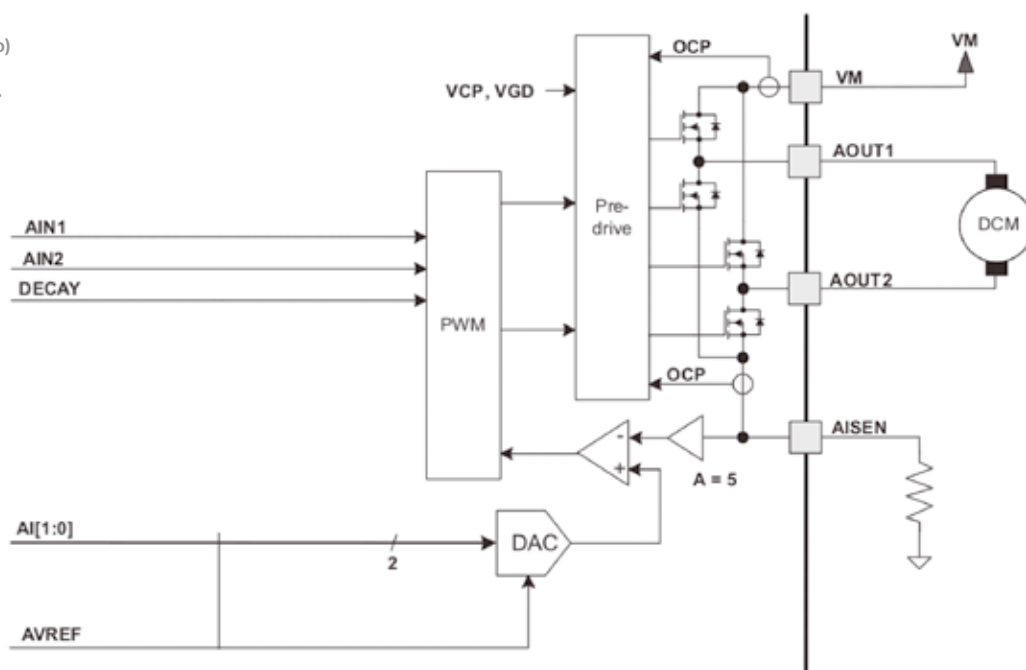


Fig. 2
Schema a blocchi (in basso)
e di connessione alla MCU
(sopra) del driver DRV8843.



MASSIMA impostata come vedremo più avanti. È comunque possibile pilotare questi due ingressi tramite un segnale PWM (e noi lo faremo, nel nostro software) per poter fornire all'avvolgimento una corrente ridotta. Applicando segnali in PWM, il funzionamento è quello descritto nella **Tabella 2**. Il 'DECAY' è il modo in cui un avvolgimento viene de-energizzato; sappiamo infatti che un'induttanza tende a mantenere la corrente in circolo quando viene disconnessa, e ci mette un po' di tempo per 'scaricarsi'.

Più velocemente si scarica ed ovviamente più rapida è la risposta del motore ai cambiamenti, al costo di maggiori disturbi impulsivi e picchi di corrente nel sistema. Nella modalità 'FAST DECAY' (decay rapido) i MOSFET del ponte vengono sconnesi e l'energia dell'avvolgimento fluisce attraverso i diodi di protezione interni ai MOSFET stessi; nella modalità 'SLOW DECAY' i mosfet cortocircuitano l'avvolgimento.

Per quanto possa sembrare poco intuitivo, un avvolgimento si scarica più velocemente tanto più è grande la resistenza utilizzata per scaricarlo, quindi inserendo i diodi di protezione nel percorso di scarica si riduce il tempo.

- **DECAY** è l'ingresso utilizzato per impostare la modalità di DECAY, appunto, quando non si utilizza il PWM. È un ingresso a 3 stati; va impostato a LOW per uno slow decay, a HIGH per un fast decay e lasciato sconnesso per un decay misto.
- **AIO ed AI1** sono due ingressi utilizzati per frazionare la corrente nell'avvolgimento; sono utilizzati come un numero a 2 bit, frazionando la corrente dal 100% (00) fino allo 0% (11) della corrente massima impostata tramite l'ingresso AVREF e la resistenza sull'ingresso AISEN.
- **AVREF** è l'ingresso di riferimento per l'impostazione della corrente dell'avvolgimento. Può essere pilotato tramite un DAC per realizzare un microstepping oppure connesso ad un riferimento fisso se si esegue il microstepping in altro modo, ad esempio tramite i pin AIN1/AIN2.
- **AISEN** è il pin destinato alla resistenza per rilevare la corrente nell'avvolgimento. La corrente MASSIMA nell'avvolgimento (ottenuta tramite AIO/AI1 entrambi a 0) è data dalla formula :

$$I_{CHOP} = \frac{V_{ref}}{5 \cdot R_{ISENSE}}$$

In MotorFish abbiamo portato tutte le connessioni del driver sul controller, in modo da poter scegliere

AIN1	AIN2	AOUT1	AOUT2
0	0	Z	Z
0	1	L	H
1	0	H	L
1	1	L	L

Tabella 1
Stato delle uscite
in funzione
di AIN1 e AIN2.

liberamente la modalità di pilotaggio del motore tramite software.

Per quanto riguarda gli ingressi analogici (**AVREF**) abbiamo sfruttato il PWM veloce ed ad alta risoluzione della PIC abbinato ad un filtro passabasso RC esterno in modo da poter generare un'ampia gamma di correnti di pilotaggio del motore; si possono notare nello schema i componenti R1/C8 ed R2/C9 che costituiscono i due filtri passa-basso necessari a ricostruire il segnale analogico di riferimento dal PWM in uscita dal controller.

L'ENCODER MAGNETICO

Finora abbiamo descritto una serie di componenti hardware che potrebbero essere relativi ad un qualsiasi controller open loop; qui di seguito descriviamo il 'pezzo forte' della scheda, ovvero l'encoder magnetico, in grado di rilevare la posizione dell'alberino del motore e trasmetterla al controller. Come componente abbiamo scelto un AM4096 di RLS, dotato di una risoluzione pari a 12 bit (4096 passi / giro). Applicato ad uno stepper con 200 passi / giro permette un microstepping massimo pari a $4096/200 = 20.48$ micropassi/passi. Può sembrare un microstepping non molto elevato, visto che in commercio si trovano facilmente controller con 1/32, 1/64 e persino 1/128 di passo, ma occorre considerare che :

- Ad un microstepping più fitto non corrisponde necessariamente una precisione più elevata, per i motivi che spiegheremo a breve

AIN1	AIN2	FUNZIONE
PWM	0	PWM IN AVANTI, DECAY RAPIDO
1	PWM	PWM IN AVANTI, DECAY LENTO
0	PWM	PWM ALL'INDIETRO, DECAY RAPIDO
PWM	1	PWM ALL'INDIETRO, DECAY LENTO

Tabella 2
Funzioni del driver
motore.



- Già un passo a 1/20 è una precisione difficilmente gestibile dalla meccanica; la 3Drag, ad esempio, ha i drivers impostati ad 1/16 di passo, e le impostazioni meccaniche di 64.25 micropassi/mm; ad un micropasso corrisponde quindi una precisione lineare pari a $1/16.25 = 15.6$ micron (millesimi di mm), ben oltre alla precisione della meccanica con i vari giochi, scorrimenti, eccetera.

A cosa servono, quindi, i microstepping elevati? Semplicemente a rendere il motore più fluido e silenzioso, diminuendo nel contempo le vibrazioni meccaniche. Aumentando i microstep, infatti, il motore si avvicina come funzionamento ad un motore brushless, che si muove senza vibrazioni. Nel nostro controller il motore è pilotato da un'onda sinusoidale sintetizzata tramite il PWM, che può avere una risoluzione elevata a piacere; il motore non si comporta quindi come un vero stepper ma come un motore brushless.

Dicevamo poco sopra che non necessariamente ad un microstepping elevato corrisponde un aumento di precisione.

Perché? Semplicemente per una caratteristica degli stepper. Il posizionamento è dato da una coppia di avvolgimenti, che attirano o respingono dei magneti permanenti nel motore. Ad un passo intero corrisponde la posizione di uno dei poli di questi magneti permanenti, e quindi ad una coppia molto elevata del motore.

A dire il vero, la coppia massima si ottiene energizzando al massimo i due avvolgimenti, in modo che il motore si posizioni in corrispondenza del punto intermedio tra due poli.

Se ci spostiamo di un passo intero ci troviamo ancora in una posizione di coppia massima; la **Fig. 3** esemplifica il funzionamento in full stepping per un motore unipolare.

Nel microstepping, invece, ci si posiziona in un punto intermedio tra i punti di coppia massima, fornendo una corrente differente ai 2 avvolgimen-

ti. In pratica, ipotizzando un microstepping con un numero infinito di passi, la corrente nei due avvolgimenti è costituita da 2 sinusoidi sfasate di 90 gradi tra loro.

Sono stati fatti vari studi in proposito, non riportati qui per brevità, che dimostrano che la coppia di mantenimento cala in modo notevole all'aumentare dei micropassi, finché non è vinta addirittura dagli attriti interni al motore stesso, senza nemmeno considerare la meccanica della stampante.

In questo caso, è possibile che il motore riceva 1-2-4-8-16 o più comandi di step senza muoversi, per poi balzare "di colpo" alla posizione richiesta, quando la coppia è sufficiente a farlo, vanificando l'apparente maggior precisione data dall'elevato numero di micropassi.

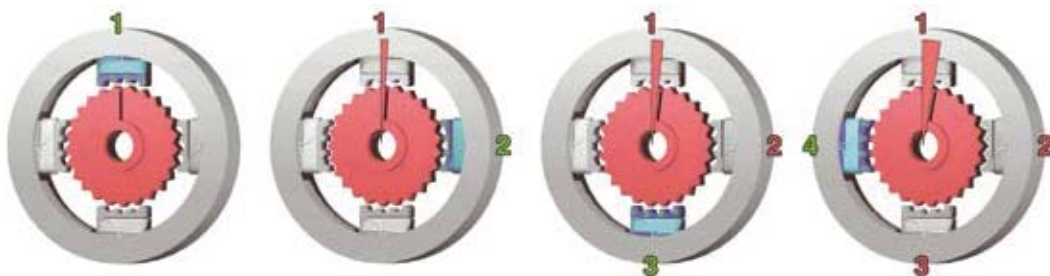
Come dicevamo sopra, il nostro encoder è in grado di risolvere 4096 passi / giro, che corrispondono ad una risoluzione lineare sulla 3Drag pari a 12.18 micron (millesimi di millimetro), decisamente più che sufficiente per qualsiasi scopo.

Un altro fatto da prendere in considerazione è costituito dalla precisione dell'encoder, da non confondere con la risoluzione, perché la risoluzione è un limite alla precisione raggiungibile da un meccanismo: per esempio con 4.096 step fisici la precisione non può superare 1/4.096 di giro. La precisione, oltre ad essere limitata dalla risoluzione dipende da altri fattori, quali la linearità dell'encoder, la ripetibilità del medesimo, eccetera. Sarà quindi sempre inferiore o, al limite, uguale alla risoluzione.

Il nostro encoder lavora con un'array di sensori ad effetto Hall ed un magnetino a polarizzazione radiale fissato sull'albero motore (come vedremo più avanti in dettaglio), ed è quindi influenzato sia dalla linearità dei sensori che dalla precisione nella realizzazione del magnete.

Il nostro software è in grado di compensare parzialmente queste non-linearità, tramite uno sketch di calibrazione, ma non può ovviare a tutti i problemi sopracitati.

Fig. 3
Schematizzazione
della modalità full
stepping.



Torniamo al nostro encoder, del quale la **Fig. 4** propone lo schema a blocchi.

Il circuito è costituito da un array di sensori ad effetto Hall in grado di rilevare l'intensità del campo magnetico in vari punti come quantità analogica, seguito da un amplificatore e da una serie di 3 blocchi:

- **Amplificatore differenziale**, utilizzato per fornire in uscita 2 sinusoidi sfasate di 90° che rappresentano la posizione dell'encoder in modo analogico
- **Interpolatore a 12 bit**, che dalle sinusoidi sfasate è in grado di ricavare un numero compreso tra 0 e 4095 corrispondente alla posizione angolare assoluta dell'encoder
- **Convertitore Digitale/Analogico da 10 bit**, che fornisce un'uscita analogica variabile a seconda della posizione

Sono presenti inoltre altri blocchi in uscita, in grado di fornire varie informazioni sulla posizione e sullo spostamento relativo del motore; abbiamo quindi un'uscita incrementale in quadratura, analoga a quella presente negli abituali encoder meccanici (**Fig. 5**). Notate che CPR sta per Counts Per Revolutions, ovvero impulsi per giro.

L'uscita **Ri** è il segnale di giro completo. Per leggere questo tipo di segnali esistono numerose librerie, utilizzate spesso per gli encoder meccanici. Abbiamo poi un'uscita in formato SSI che restituisce la posizione assoluta dell'encoder in formato digitale (**Fig. 6**).

Quest'uscita fornisce un numero binario lungo al massimo 12 bit (a seconda della risoluzione scelta) in formato seriale con il relativo clock; è leggibile tramite l'interfaccia SPI oppure tramite software. Abbiamo poi un'uscita a tre fasi di tipo UVW, non utilizzata nel nostro progetto e un'uscita lineare a dente di sega, anch'essa non di nostro interesse

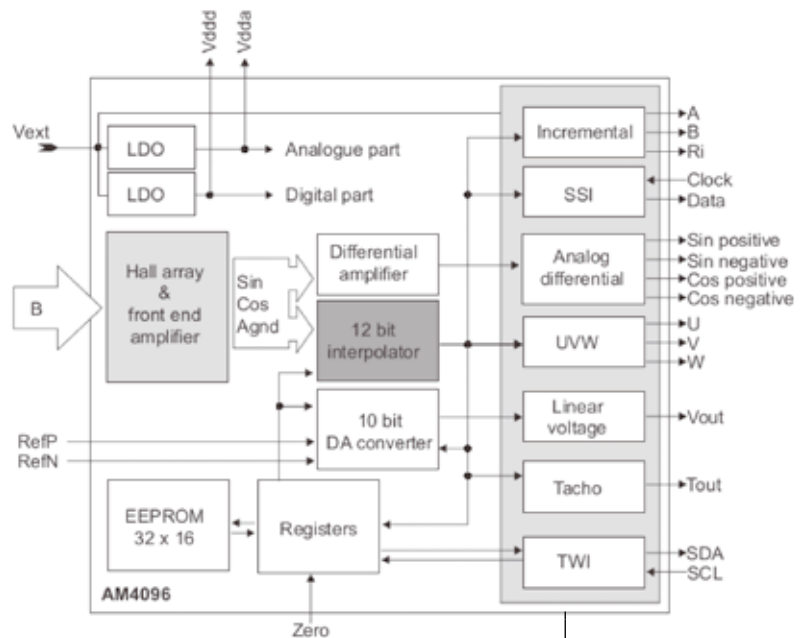


Fig. 4
Schema a blocchi dell'encoder.

(di questa sappiate che si può scegliere il periodo da 1 dente/giro fino a 8 denti/giro. Anche l'uscita tachimetrica, per rilevare la velocità di rotazione, non viene gestita nel nostro progetto. Infine troviamo quattro uscite analogiche sinusoidali, volute principalmente per i test, che sono in pratica le uscite dirette dell'array di sensori ad effetto di Hall (**Fig. 7**).

Come si può notare, l'encoder dispone di un notevole numero di funzionalità, e tutte sono programmabili tramite un'apposita serie di registri accessibili attraverso l'interfaccia I²C. Vedremo a breve, nella sezione relativa al software, come gestire questo encoder tramite il nostro controller.

REALIZZAZIONE PRATICA

MotorFish, per poter essere posizionato sul retro di un piccolo stepper in formato NEMA17, ha una costruzione piuttosto compatta; il PCB è a due facce, entrambe popolate da componenti. Il montaggio in sè, una volta disponibile un circuito

Fig. 5
Segnale incrementale all'uscita dell'encoder.

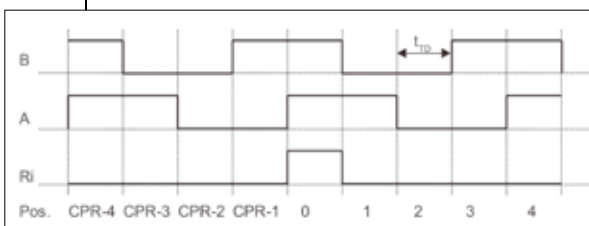
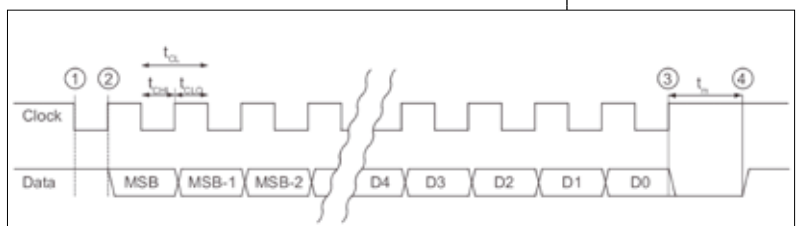
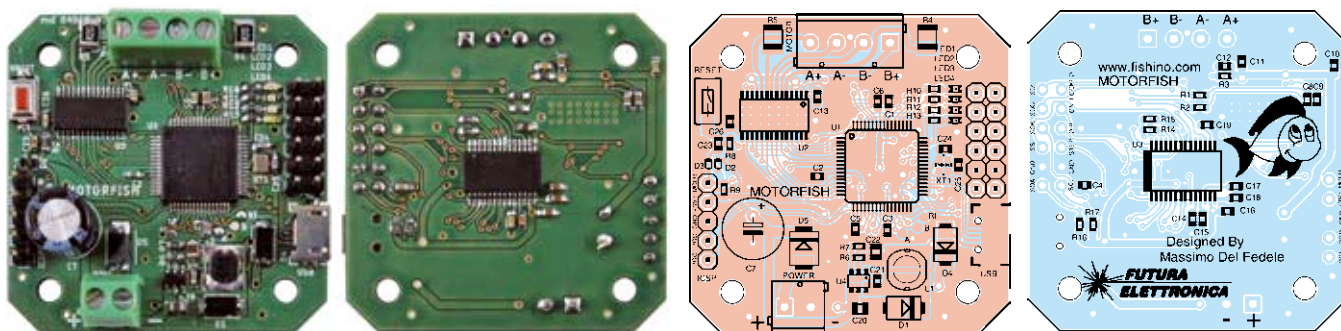


Fig. 6
Segnale dell'uscita SSI.





Elenco Componenti:

R1, R2, R8: 1 kohm (0603)
 R3: 1 Mohm (0603)
 R4, R5: 0,2 ohm (1210)
 R6: 1,02 kohm (0603)
 R7: 3,4 kohm (0603)
 R9: 10 kohm (0603)
 R10, R11, R12, R13: 470 ohm (0603)
 R14, R15, R16, R17: 2,2 kohm (0603)
 C1: 4,7 μ F ceramico (0603)
 C2, C3, C4, C5, C6, C8, C9, C10,
 C11, C12: 100 nF ceramico (0603)
 C7 100 μ F 50 VL elettrolitico
 C13, C14, C15 10 nF ceramico (0603)
 C16, C17, C18 10 μ F ceramico (0603)

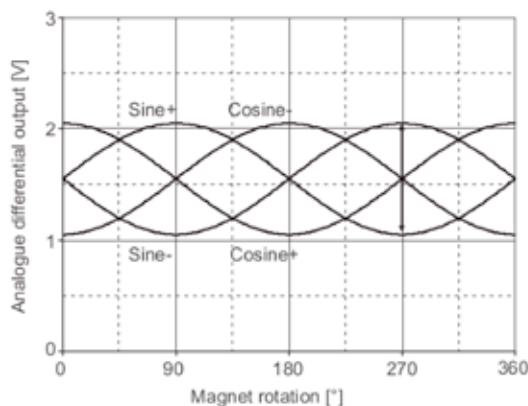
C19, C23 1 μ F ceramico (0603)
 C20 2,2 μ F 50 VL ceramico (0805)
 C21 150 nF 50 VL ceramico (0603)
 C22 47 μ F 5 VL ceramico (0805)
 C24, C25 22 pF ceramico (0603)
 D1, D4 SS36SMA
 D2, D3 RB521S
 D5 SS56SMB
 L1 12 μ H (CD54)
 LED1 LED giallo (0603)
 LED2 LED blu (0603)
 LED3 LED rosso (0603)
 LED4 LED bianco (0603)
 U1 PIC32MX250F256HI/PT (MF1439)

U2 DRV8843
 U3 AM4096
 U4 LMR14206
 RESET Microswitch
 XT1 Quarzo 20 MHz

Varie:

- Connettore microUSB
- Strip maschio 5 vie
- Strip maschio 6 vie (2 pz.)
- Morsetto 4 vie 3.5mm estraibile
- Morsetto 2 vie 3.5mm estraibile
- Circuito stampato S1439

Fig. 7
 Segnali
 delle uscite
 sinusoidali.



stampato di buona qualità, non presenta eccessive difficoltà; le problematiche principali sono il posizionamento preciso dell'encoder (che dev'essere posizionato con molta precisione, in modo che l'area sensibile sia esattamente al centro della scheda) e la saldatura del pad termico sotto al driver, senza la quale è facile che il driver stesso vada in arresto termico durante un uso intensivo a correnti elevate.

La seconda operazione è fattibile solo tramite un fornello oppure una stazione ad aria calda ed una buona mano; il resto è ordinaria amministrazione. Come solitamente facciamo nei montaggi con

componenti SMD, consigliamo di iniziare dagli elementi a basso profilo e con più difficoltà di saldatura, per non trovarsi dopo a dover "girare attorno" a componenti ingombranti. Quindi, iniziate dal driver, che richiede una buona dose di perizia per saldare il pad sotto al corpo; successivamente procedete con il controller (la PIC), in modo da rendere "stabile" il PCB una volta capovolto. Come terzo componente, montate l'encoder (sul lato inferiore), posizionandolo con estrema cura. Dalla sua perfetta centratura dipenderà infatti in gran parte la precisione. Proseguite quindi con il piccolo regolatore switching, il quarzo, resistenze e condensatori SMD, i LED, l'induttanza (anche questa è consigliabile saldarla ad aria calda, anche se è evitabile), i due grossi diodi SMD, il connettore USB ed il pulsante di reset. Per ultimo montate le due morsettiere e il condensatore elettrolitico.

I COLLEGAMENTI

MotorFish dispone di due morsettiere, una a 4 poli destinata ai due avvolgimenti del motore e una bipolare per l'alimentazione. Il collegamento del motore è rappresentato tramite le lettere **A+/A-** e **B+/B-**; corrispondono al primo ed al secondo avvolgimento del motore. La polarità è importante, anche se è praticamente impossibile determinare

quella corretta senza effettuare una prova. Misurate quindi con un Ohmetro i fili del motore, in modo da individuare ciascuna coppia (i fili dello stesso avvolgimento presentano resistenza pressochè nulla tra loro) e collegateli al connettore, senza badare alla polarità (+ o -). Quando eseguirete il collaudo con lo sketch **MotorFishClosedLoop** se noterete che il motore invece di fermarsi su un punto inizia a correre all'impazzata invertite una sola coppia, ad esempio scambiate tra di loro i fili su A+ ed A-.

A differenza di uno stepper normale, nel nostro non è possibile scambiare i fili per invertire la direzione del motore, visto che la direzione dell' Encoder è prefissata e deve corrispondere con quella del motore stesso. Poichè a volte è necessario scambiare la direzione di rotazione, ad esempio succede nella 3Drag dovendo ribaltare uno dei motori per motivi di spazio, abbiamo previsto una variabile nello sketch per gestirla.

Al connettore di alimentazione va applicata una tensione compresa tra circa 8,5 e 45 V; maggiore è, meglio è, visto che ci sarà un minore assorbimento di corrente (il funzionamento è switching!) ed in contemporanea migliori prestazioni del motore. Nella 3Drag abbiamo comunque utilizzato la tensione standard di 15 Volt ed il tutto funziona perfettamente.

Attenzione alla polarità, chiaramente indicata sul PCB. Invertendola non "esplode" comunque nulla, abbiamo previsto sulla scheda un diodo di protezione contro le inversioni di polarità.

Passiamo ora al connettore che porta i segnali di controllo; poichè le scritte sono sul lato inferiore del PCB, per motivi di spazio, mostriamo la vista da sotto in **Fig. 8**. Ricordatevi quando effettuerete i collegamenti!

Come potete notare, MotorFish dispone di una buona dotazione di interfacce, gestibili tramite il software; è quindi possibile controllarlo, oltre che attraverso l'interfaccia USB, tramite una porta I2C o una porta SPI.

Dispone inoltre dei classici input di ON/DIR/STEP tipici dei controller tradizionali, e che ne rendono possibile la sostituzione diretta con MotorFish.h

- **SS, SCK, SDO e SDI** sono i pin relativi all'interfaccia **SPI**
- **SDA e SCL** sono i pin relativi all'interfaccia **I2C**
- **STEP, DIR e ON** sono i pin relativi all'interfaccia stepper tradizionale
- **LOCKED** è un'uscita di "aggancio" che indica quando il motore ha raggiunto la posizione prescelta; può essere utilizzato come segnale di



Fig. 8
Piazzole dei
collegamenti di
segnale.

errore o comunque per comunicare al software della stampante eventuali problemi.

Va comunque detto che TUTTI gli I/O disponibili sono programmabili da software, quindi nulla vieta di utilizzarli per altro. MotorFish si programma esattamente come un Arduino (o, più precisamente, come un Fishino32), con tutte le possibilità del caso. Resta da vedere, per ultimo ma non come importanza, il connettore previsto per il caricamento del bootloader, ossia il connettore siglato ISP; quest'ultimo è stato previsto per essere utilizzato direttamente con un programmatore di PIC, ad esempio il PicKit3, in abbinamento con il software MPLAB IDE della Microchip. Le uscite PGC e PGD sono comunque utilizzabili come due I/O aggiuntivi, quando non si deve ricaricare il bootloader sulla scheda. Se acquisterete il MotorFish già montato il bootloader sarà comunque precaricato e quindi potrete utilizzare la scheda con l' IDE di Arduino direttamente, programmandola via USB.

CONCLUSIONI

Bene, per questa puntata ci fermiamo qui; riprenderemo il discorso nella prossima ed ultima spiegando il caricamento del bootloader e del firmware, nonché descrivendo il firmware stesso che permette il funzionamento di Motorfish. Vedremo anche come configurare questo potente e versatile controller grazie al menu accessibile collegandosi con il terminale seriale alla scheda. ■



Cosa occorre?

Il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. Il controller con retroazione per motori passo-passo (cod. MOTORFISH) è in vendita a Euro 55,00, il motore NEMA17 (cod. STEPMOT03) è disponibile a Euro 18,00.

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

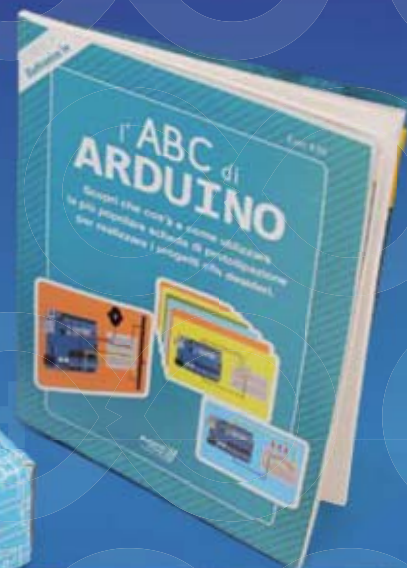


Programma e realizza applicazioni elettroniche con ARDUINO!

IL PIÙ
COMPLETO!

SET PER ARDUINO UNO REV3

Set contenente tutto il necessario per sviluppare applicazioni con ARDUINO UNO Rev3. Il kit comprende: il libro "l'ABC di ARDUINO", la board Arduino UNO Rev3, shield di prototipazione, breadboard, connettori, contenitore plastico, LED, buzzer, pulsanti, display vari, interruttori, fotoresistenze, potenziometro, jumper, sensori, ricevitore infrarossi, telecomando IR, resistenze, cavetti, display LCD 16 caratteri, servo, motore passo-passo, scheda driver, chip 74HC595.



€ 65,00

Cod. **ARDUKITV6**

Utilizzalo anche con il kit di 37 sensori

Moduli con sensori per Arduino, Raspberry, robotica ecc.

La confezione contiene i seguenti moduli:

- Joystick da C.S.
- Sensore di fiamma
- Pulsazioni cardiache
- LED e interruttore al mercurio
- Sensore effetto di Hall
- Relè da 5 V
- Sensore effetto Hall lineare
- LED RGB SMD
- LED cambia colore
- Sensore di Tilt
- Sensore di temperatura DS18B20
- Rilevatore di suono con microfono grande
- Touch sensor
- LED bicolore 5 mm
- LASER
- Sensore di Tilt a inclinazione
- Sensore di temperatura analogico
- Rilevatore di suono con microfono piccolo
- Sensore di temperatura digitale
- LED bicolore 3 mm
- Pulsante N.A.
- Fotoresistenza
- LED trasmettitore IR
- Sensore di linea bianca e nera
- Buzzer con elettronica
- Interruttore reed e interfaccia digitale
- Interruttore a vibrazione
- Sensore di temperatura e umidità
- Interruttore reed
- Ricevitore IR a 3 pin
- Rilevatore di ostacoli
- Buzzer senza elettronica
- Encoder rotativo
- Sensore effetto di Hall analogico
- Sensore urto
- Mini photo interruttore a forcella

€ 29,90

Cod. **SENSORKIT37**



IRRIGAZIONE A CONTROLLO BT

di **ANTONIO VOLPE**



Analizziamo il software del progetto e l'interfaccia utente, che è costituita da un'App sviluppata con Android Studio.

Una mappa messaggi ben definita determina come app e firmware devono scambiarsi le informazioni. Seconda e ultima puntata.

Nella puntata precedente abbiamo descritto come configurare e assemblare in pochi passi l'hardware del nostro sistema di irrigazione; adesso dobbiamo "attivarlo" ossia implementare il software che gli consente di svolgere il compito cui lo abbiamo destinato. Nel nostro caso la parte software consta di due parti: il firmware da caricare su Mercury e l'app da installare sullo smartphone, che costituirà l'interfaccia utente dalla quale gestire il sistema. L'implementazione del software è necessaria per determinare quali azioni deve compiere il nostro sistema, quando e sotto quali condizioni, per essere sempre informati dello stato corrente dell'irrigazione o per comandarla su richiesta. Il lavoro di sviluppo del software è semplificato dall'utilizzo del Framework di Mercury System, il cui livello di astrazione ci permette di focalizzare la nostra attenzione sullo sviluppo dell'applicazione senza preoccuparci di configurare l'hardware. Quanto all'app, verrà sviluppata con Android Studio.

Una mappa messaggi ben definita determinerà come App e firmware dovranno scambiarsi le informazioni; analizziamo dunque per prima proprio tale mappa, così da aver poi chiaro il perché firmware e App funzionano come vedrete.

MAPPA MESSAGGI

Guardando la **Fig. 1** si vede che il canale di comunicazione tra i due software (l'App e il firmware) è determinato dal blocco Connectivity; nel nostro caso la Connectivity scelta è quella Bluetooth, ma potrebbe essere sostituita anche con altre tipologie. Possiamo pensare ai blocchi **Mercury System**, che conterrà un software sviluppato in **C**, e **Smartphone**, su cui verrà installata un'app sviluppata in **Java** (Android Studio), come a due persone che parlano lingue diverse: per metterle in comunicazione e fare in modo che si capiscano dobbiamo definire un set di informazioni chiaro per entrambe, ossia una **Mappa Messaggi**.

Nella documentazione a corredo è presente la mappa messaggi descritta in modo dettagliato, ma illustriamo brevemente sotto, quali sono i messaggi disponibili e il loro significato/utilizzo:

- **SYS_TIME_CFG** serve a configurare la data corrente: l'App invia questo messaggio ogni volta che effettua una connessione al sistema;
- **IRR_TIME_CFG** configura la schedulazione delle attività delle singole elettrovalvole: tramite questo messaggio vengono trasmesse o ricevute le programmazioni da salvare o esistenti nel sistema;
- **IRR_TIME_RESET** resetta l'intero programma;
- **SYS_ZONE_CFG** configura il numero di zone scelte dall'utente per un massimo di 4;
- **SYS_ZONE_CMD** consente l'attivazione manuale/automatica delle elettrovalvole;
- **SYS_STS_REQ** comunica quali sono le elettrovalvole attive;
- **REM_IRR_TIME** indica il tempo mancante per terminare l'attività in corso per ogni elettrovalvola;
- **IRR_TX_TIME_CFG** invia all'App tutte le irrigazioni schedulate memorizzate.

Una volta deciso cosa devono comunicarsi Mercury

e lo Smartphone, non ci resta che passare alla descrizione del firmware e dell'app. Come spiegato in precedenza il software si compone di due parti, un software per Mercury e un'app per lo smartphone. Descriviamo il funzionamento del sistema partendo dalla descrizione dell'app, che ad alto livello ci fa intuire in maniera più semplice come funziona il sistema, successivamente passiamo alla descrizione del firmware.

APP ANDROID

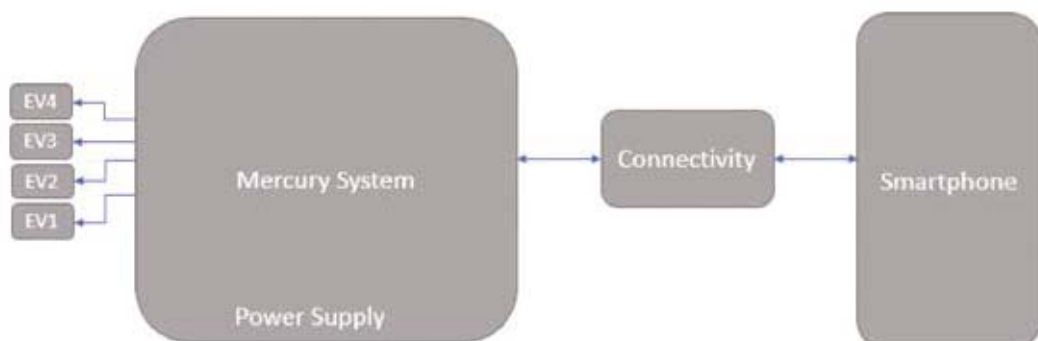
L'App Android realizzata in Java con Android Studio è costituita dalle finestre riportate sotto nei layout di transizione.

Dalla Main Window, premendo il pulsante **BT** si apre la finestra **Devices List** che ci permette di effettuare il pairing tra il dispositivo su cui è installata l'app e l'Irrigation System mostrandoci la lista dei dispositivi disponibili. In questo caso selezioniamo il dispositivo HC-05, la finestra si chiude tornando alla Main Window e una volta avvenuto il pairing il pulsante BT si colora di verde: un Toast Message ci comunicherà se la connessione è avvenuta con successo o meno. Vedremo a questo punto che sul display di Irrigation System, verrà impostato il giorno e l'orario corrente. Sulla Main Window possiamo vedere che per ciascuna elettrovalvola sono disponibili le seguenti informazioni:

- un'etichetta che si colora di verde quando l'elettrovalvola corrispondente è in funzione;
- il tempo di funzionamento totale rispetto al tempo programmato;
- un pulsante per azionare manualmente l'elettrovalvola se configurata. Nel caso in cui l'elettrovalvola non sia configurata il pulsante non è cliccabile.

Una volta effettuata la connessione ad Irrigation System, il pulsante **Impostazioni** diventa cliccabile,

Fig. 1
Schema a blocchi del sistema.



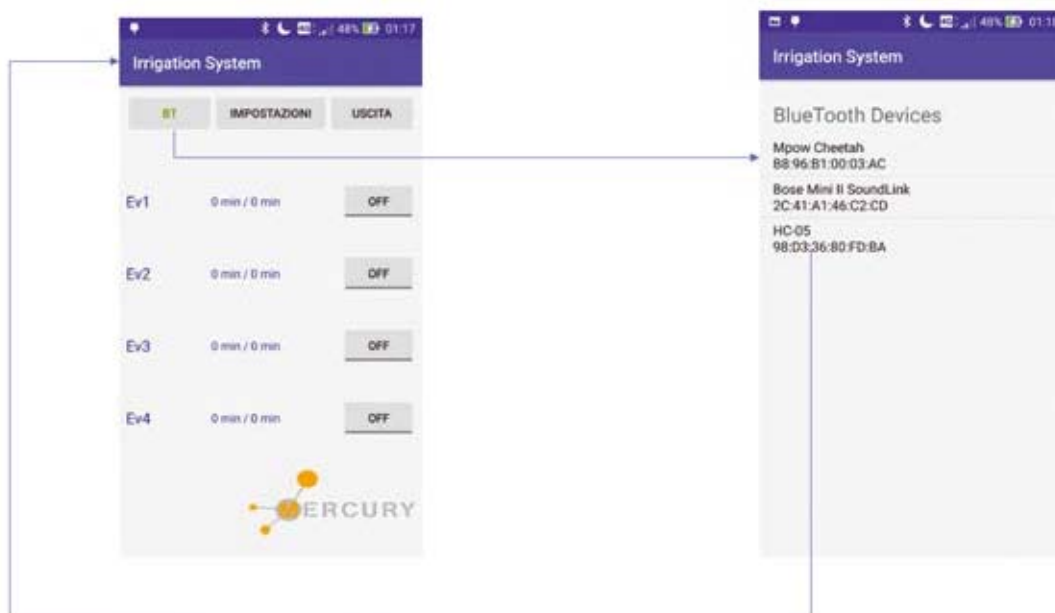


Fig. 2
Transizione da Main Window a Devices List Window.

possiamo quindi passare alla configurazione del sistema. Una volta aperta la finestra Configuration Window, bisogna abilitare il Setup e configurare le elettrovalvole che intendiamo utilizzare. Se non vogliamo programmare azionamenti automatici, possiamo salvare e tornare alla Main Window, altrimenti compilando i campi presenti nella Configuration Window, possiamo aggiungere le programmazioni che ci servono per un massimo di 28 a zona. Una volta terminato l'inserimento

delle configurazioni, bisogna premere il tasto **Salva**: le informazioni verranno trasferite ad Irrigation System. A questo punto Irrigation System eliminerà tutte le configurazioni precedentemente salvate sostituendole con le nuove.

SOFTWARE MERCURY: OVERVIEW

Per la realizzazione del firmware ho utilizzato la versione 1.0.0 del Framework Mercury la cui architettura è descritta in **Fig. 4**. Per scrivere il sof-

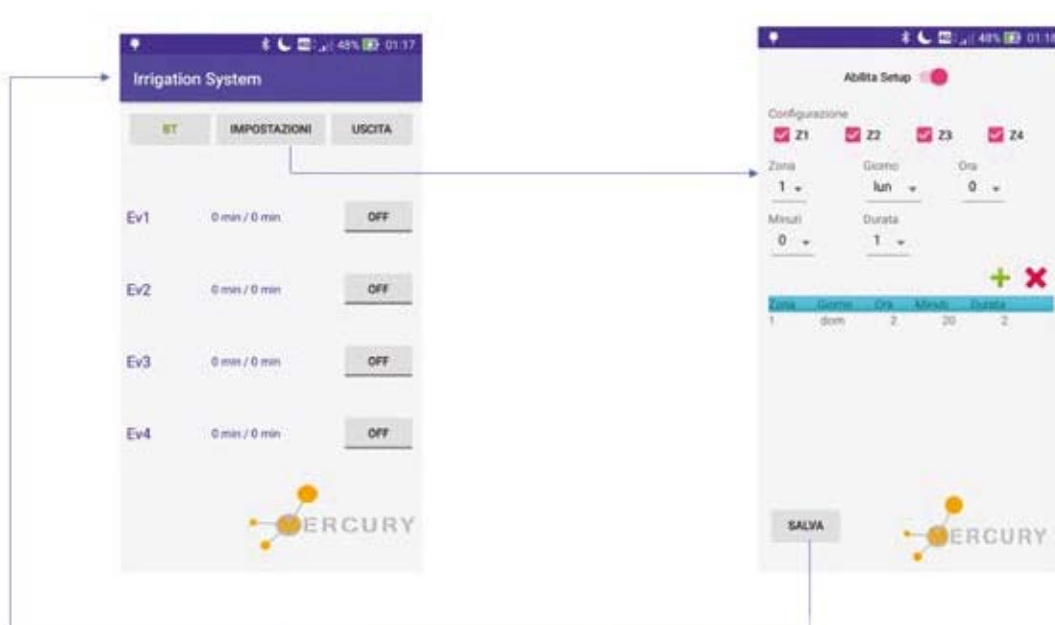
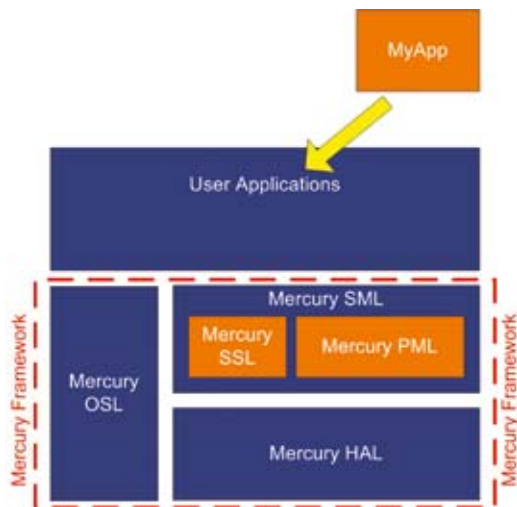


Fig. 3
Transizione da Main Window a Configuration Window.

➔ Fig. 4
Framework
Mercury.



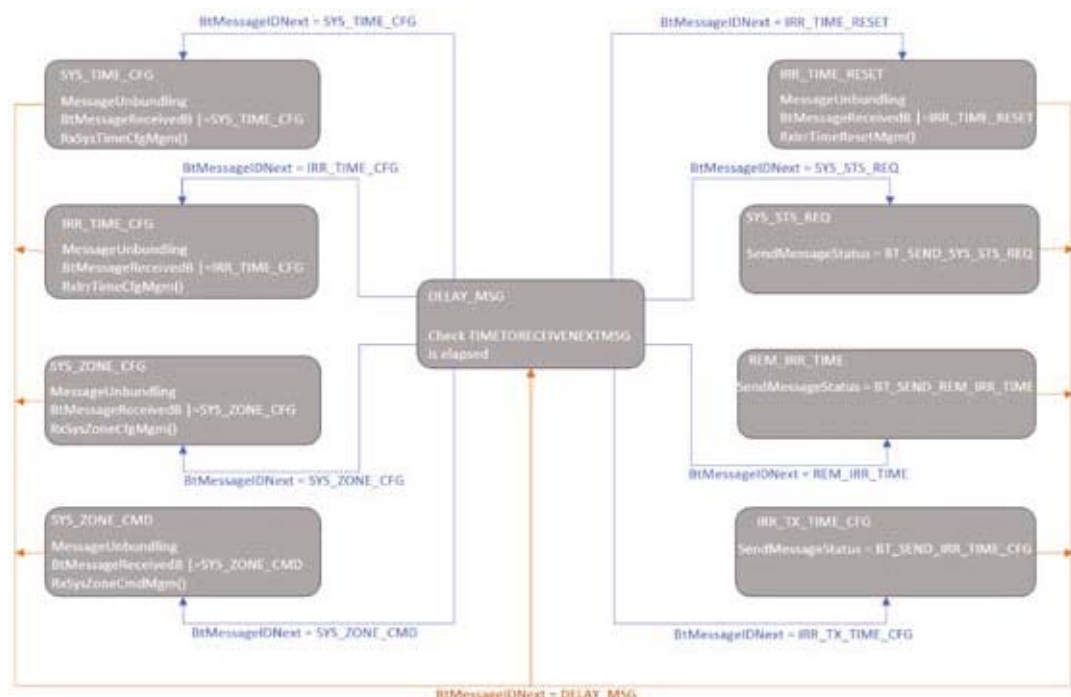
tware non dobbiamo preoccuparci di come è fatto il microcontrollore e tutto l'hardware che abbiamo scelto di utilizzare, Mercury ci fornisce tutti i servizi necessari e le API per focalizzare la nostra attività sullo sviluppo della User Application. Nell'ultimo articolo su Mercury System pubblicato sul numero 234 di Elettronica In, abbiamo visto come possiamo aggiungere al sistema operativo un'applicazione semplicemente aggiornandola nella schedule-table così come indicato nel **Listato 1**. Chiamiamo la nostra applicazione

ApplIrrigationSys_Task e impostiamo un periodo di 10 ms definendolo in **APP_TASK_PERIOD_MS**.

- **ApplIrrigationSys_Task**, è un'applicazione costituita da vari manager i quali si occupano ciascuno di funzioni specifiche:
- **BTReceiveSendMgr**: invia e riceve messaggi Bluetooth.
- **IrrigationMgr**: verifica se ci sono irrigazioni da memorizzare, se occorre accendere/spegnere le elettrovalvole in modalità automatica o manuale.
- **SystemTimeConfigMgr**: memorizza la data/ora corrente se richiesto dall'App
- **SystemTimeMgr**: si occupa di gestire le informazioni relative al tempo: giorno, ora, minuti, secondi
- **LcdMgr**: aggiorna le informazioni sul tempo indicate sul display LCD

Abbiamo quindi visto che ogni manager ha un compito ben preciso: ricezione/invio dei messaggi, gestione delle irrigazioni, gestione del tempo e, infine, gestione del display della Expansion Board. Dal **Listato 2** è possibile vedere che ad alto livello la **ApplIrrigationSys_Task** è molto semplice, infatti, ogni 10 ms controlliamo quali manager devono essere chiamati. Stabilito ciò, andiamo a vedere e spiegare in che modo lavorano i vari manager.

➔ Fig. 5
Schema a
blocchi della
macchina a stati
per la ricezione
di messaggi
Bluetooth.



↓ Listato 1

```
/* ----- Application Tasks ----- */
/* ----- */
/* Task                State          Counter          Timeout          Options */
/* ----- */
#if (APP_TASK_STS == STD_ON)
{AppIrrigationSys_Task, TaskIdle, COUNTER_INIT, APP_TASK_PERIOD_MS, NO_OPTIONS},
#endif
/* ----- */
};
```

I MANAGER

I manager sono costituiti da state machine, ognuna con un compito ben preciso. Le state machine vengono eseguite ogni volta che **AppIrrigation-Sys_Task** ne decide l'esecuzione. Descriviamo i vari manager e le loro state machine.

BTReceiveSendMgr

Invia e riceve messaggi Bluetooth gestiti su base evento, ad esempio, quando l'app lo richiede o quando il firmware attua qualche irrigazione; questo viene fatto per creare traffico sul canale Bluetooth solo quando serve. Mercury provvede, infatti, a inviare messaggi solo allo scatenarsi di un determinato evento, mentre l'app se necessita di specifiche informazioni, invia una richiesta di invio

messaggio il quale conterrà ciò che ha richiesto. Ad esempio quando l'app effettua una connessione BT chiede a Mercury quali sono le elettrovalvole attive. Per utilizzare il modem Bluetooth bisogna impostare il relativo compile switch nel file `sys_cfg.h`:

```
#define MODEM_USED          BT_MDM
```

A questo punto il nostro sistema è pronto per ricevere ed inviare messaggi. Dal punto di vista del Firmware, quindi lato Mercury, tutti i messaggi previsti in **mappa messaggi** possono essere ricevuti, mentre sono solo tre i messaggi che possono essere trasmessi:

- **BT_SEND_SYS_STS_REQ**
- **BT_SEND_REM_IRR_TIME**
- **BT_SEND_IRR_TIME_CFG**

↓ Listato 2

```
void AppIrrigationSys_Task (uint8_t Options)
{
    static uint8_t IrrigationMgrTm = 0;
    static uint8_t LcdMgrTm = 0;

    BTReceiveSendMgr();

    if(IrrigationMgrTm >= IrrigationTmP)
    {
        IrrigationMgrTm = 0;
        IrrigationMgr();
    }

    SystemTimeConfigMgr();
    SystemTimeMgr();

    #if (DISPLAY_MGM_ENABLE == STD_ON)
    if(LcdMgrTm >= LcdMgrTmP)
    {
        LcdMgrTm = 0;
        LcdMgr();
    }
    #endif

    LcdMgrTm++;
    IrrigationMgrTm++;
}
```

Delle due funzioni chiamate all'interno del Manager, la prima si occupa della ricezione e dello spaccettamento dei messaggi, mentre la seconda impacchetta i messaggi e li spedisce:

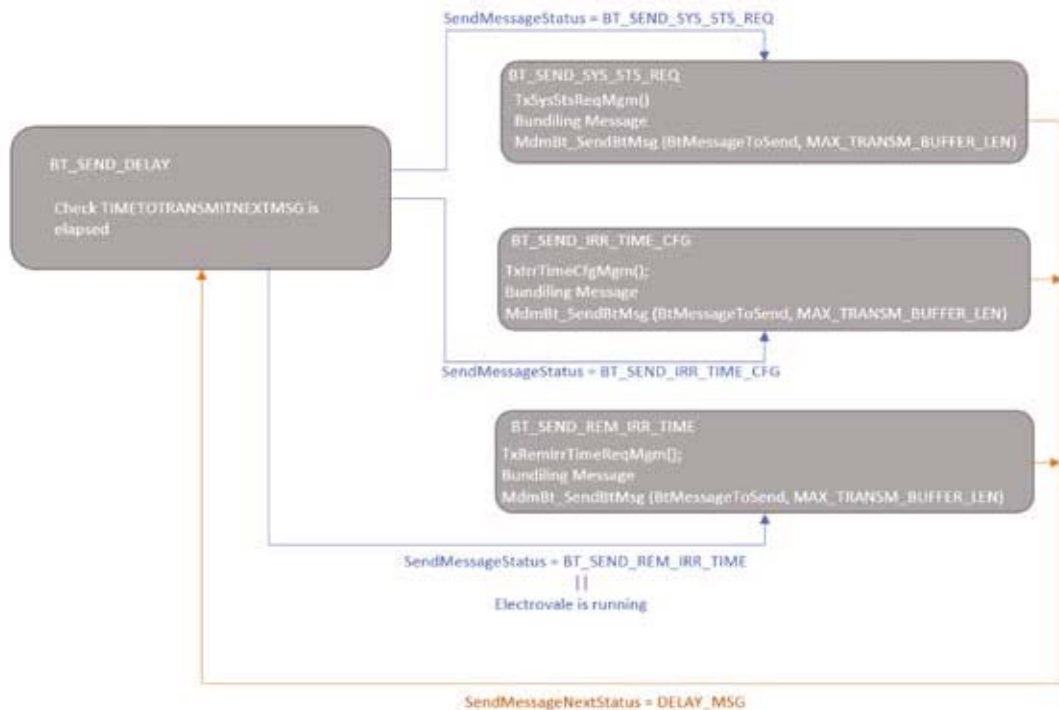
```
void BTReceiveSendMgr(void)
{
    BtReceiveMessages();
    BtTransmitMessages();
}
```

Nella **Fig. 5** vediamo uno schema a blocchi della macchina a stati che gestisce la ricezione dei messaggi BT. Ad ogni esecuzione del manager Bluetooth la funzione **BtReceiveMessages()** chiama l'API **MdmBt_ReceiveBtMsg(BtMessageReceived, &Lenght)** in cui **BtMessageReceived** è il buffer ricevuto e **Lenght** la sua lunghezza.

Se l'API restituisce **BtMsg_Received** significa che è stato ricevuto un messaggio, il primo byte di **BtMessageReceived** contiene l'identificativo del



Fig. 6
Schema a Blocchi
macchina a stati
per la trasmissione
di messaggi
Bluetooth.



messaggio che viene memorizzato in **BtMessageID**; se questo fa parte del set di messaggi che il sistema elabora, la macchina a stati schematizzata in **Fig. 5** gestirà opportunamente il messaggio che viene impacchettato nella **BtMessageMatrix**. Ogni messaggio ricevuto, prevede una funzione dedicata che ne elabora le informazioni, secondo quanto descritto qui di seguito:

- l'inizio dell'elaborazione di un messaggio è determinato dal fatto che viene **settata** nella maschera a bit **BtMessageReceivedB** l'identificativo del messaggio ad esempio **BtMessage-**

ReceivedB |= SYS_TIME_CFG.

- La fine dell'elaborazione di un messaggio è determinata dal fatto che viene **re-settata** nella maschera a bit **BtMessageReceivedB** l'identificativo del messaggio e la fase di ricezione può dirsi completata, ad esempio **BtMessageReceivedB &= ~SYS_TIME_CFG.**

Nel **Listato 3** proponiamo parte del codice implementato per la ricezione del messaggio **SYS_TIME_CFG**. Nel **Listato 4**, invece, è riportato il codice per la funzione **RxSysTimeCfgMgm**.

Per l'invio dei messaggi invece è stata implementa-

↓ Listato 3

```

case SYS_TIME_CFG :
    BtMessageMatrix[SYS_TIME_CFG_IDX].id = BtMessageID;
    BtMessageMatrix[SYS_TIME_CFG_IDX].bytenumber = Lenght;
    for(MessageBufferIdx = 0; MessageBufferIdx < BtMessageMatrix[SYS_TIME_CFG_IDX].bytenumber;
    MessageBufferIdx++)
    {
        BtMessageMatrix[SYS_TIME_CFG_IDX].buffer[MessageBufferIdx] =
        BtMessageReceived[MessageBufferIdx + 1];
    }
    BtMessageReceivedB |= SYS_TIME_CFG;
    BtMessageID = DELAY_MSG;
    BtMessageIDNext = DELAY_MSG;

    RxSysTimeCfgMgm();
    break;
  
```

Listato 4

```
if((BtMessageReceivedB & SYS_TIME_CFG) != 0)
{
    ConfigSystemtime.days    = BtMessageMatrix[SYS_TIME_CFG_IDX].buffer[BYTE_0];
    ConfigSystemtime.hours   = BtMessageMatrix[SYS_TIME_CFG_IDX].buffer[BYTE_1];
    ConfigSystemtime.minutes = BtMessageMatrix[SYS_TIME_CFG_IDX].buffer[BYTE_2];
    ConfigSystemtime.seconds = BtMessageMatrix[SYS_TIME_CFG_IDX].buffer[BYTE_3];
    CfgSysTimeEn = TRUE;    //abilito la configurazione
    BtMessageReceivedB &= ~SYS_TIME_CFG;
}
```

ta la funzione **BtTransmitMessages()**. La trasmissione dei messaggi, gestita dalla macchina a stati visibile nella **Fig. 6**, è decisa di volta in volta dai vari manager o dall'App, che può richiedere la trasmissione di un particolare messaggio. I messaggi che devono essere trasmessi, vengono prima impacchettati nella **BtMessageMatrix** da una funzione dedicata, a questo punto viene settato il flag **BtMessageTransmittedB** che indica che il messaggio è pronto per essere trasmesso, viene poi composto il buffer **BtMessageToSend** che viene inviato chiamando l'API **MdmBt_SendBtMsg (BtMessageToSend, MAX_TRANSM_BUFFER_LEN)**. In fine viene resettato il flag **BtMessageTransmittedB** per indicare che la trasmissione si è conclusa. Nel **Listato 5** trovate il codice implementato per l'invio del messaggio **BT_SEND_SYS_STS_REQ**.

IrrigationMgr

È costituito da una macchina a stati **EvMgmStatusV** che al primo avvio si porta nello stato di **INIT** in cui un'ulteriore macchina a stati **InitMgm** resetta tutte le programmazioni ai valori predefiniti, per poi passare allo stato di inizializzazione delle elettrovalvole in modo da spegnerle nel caso fossero rimaste accese.

Terminata la fase di inizializzazione, **EvMgmStatusV** si porta nello stato di **EV_IDLE** in cui verifica minuto per minuto se ci sono irrigazioni da schedulare in maniera automatica e verifica se ci sono elettrovalvole da comandare in modalità manuale. Osservando lo schema a blocchi di **Fig. 7**, quando viene avviata un'irrigazione automatica, **EvMgmStatusV** partendo dallo stato **EV_IDLE**, si porta nello stato **EV_RUN_AUTO_MODE** e ci

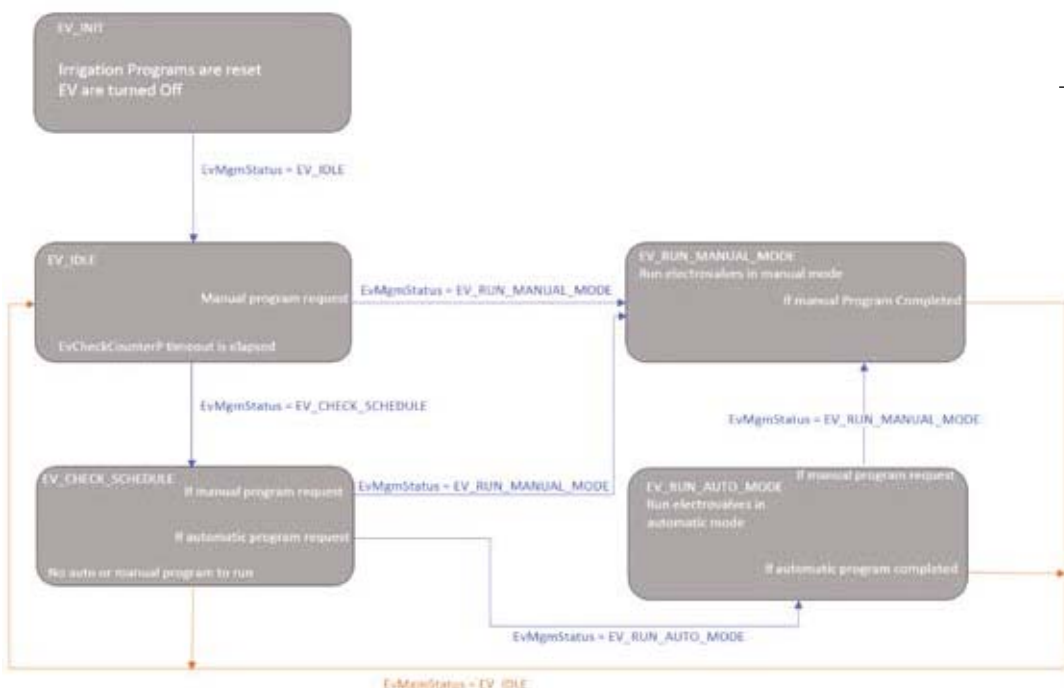


Fig. 7
Schema a Blocchi
IrrigationMgr.

Listato 5

```

case BT_SEND_SYS_STS_REQ :

    TxSysStsReqMgm();

    if((BtMessageTransmittedB & SYS_STS_REQ) != 0)
    {
        SendMessagePreamble = 0;
        BtMessageToSend[SendMessagePreamble++] = BtMessageMatrix[SYS_STS_REQ_IDX].id;

        for(MessageBufferIdx = SendMessagePreamble; MessageBufferIdx < MAX_TRANSM_BUFFER_LEN;
        MessageBufferIdx++)
        {
            if(MessageBufferIdx < BtMessageMatrix[SYS_STS_REQ_IDX].bytenumber)
            {
                BtMessageToSend[MessageBufferIdx] = BtMessageMatrix[SYS_STS_REQ_IDX].
buffer[MessageBufferIdx - SendMessagePreamble];
            }
            else
            {
                BtMessageToSend[MessageBufferIdx] = 0;
            }
        }
        MdmBt_SendBtMsg (BtMessageToSend, MAX_TRANSM_BUFFER_LEN);
        BtMessageTransmittedB &= ~SYS_STS_REQ;
    }
    SendMessageStatus = BT_SEND_DELAY;
    SendMessageNextStatus = BT_SEND_DELAY;
    break;

qui invece la funzione TxSysStsReqMgm:
void TxSysStsReqMgm(void)
{
    uint8_t ZoneIdx;

    BtMessageMatrix[SYS_STS_REQ_IDX].id = SYS_STS_REQ;
    BtMessageMatrix[SYS_STS_REQ_IDX].bytenumber = SYS_STS_REQ_BYTE_LEN;
    BtMessageMatrix[SYS_STS_REQ_IDX].period = SYS_STS_REQ_PRD;
    BtMessageMatrix[SYS_STS_REQ_IDX].type = BT_MESSAGE_TX;

    BtMessageMatrix[SYS_STS_REQ_IDX].buffer[BYTE_0] = 0;
    BtMessageMatrix[SYS_STS_REQ_IDX].buffer[BYTE_1] = 0;
    BtMessageMatrix[SYS_STS_REQ_IDX].buffer[BYTE_2] = 0;

    for(ZoneIdx = 0; ZoneIdx < MAX_NUM_ELECTROVALVES; ZoneIdx++)
    {
        BtMessageMatrix[SYS_STS_REQ_IDX].buffer[BYTE_0] |= IrrigationEv[ZoneIdx].config << ZoneIdx;
        BtMessageMatrix[SYS_STS_REQ_IDX].buffer[BYTE_1] |= ((IrrigationEv[ZoneIdx].activationstatus ||
EvCmdAutoMode[ZoneIdx]) << ZoneIdx);
        BtMessageMatrix[SYS_STS_REQ_IDX].buffer[BYTE_2] |= (IrrigationEv[ZoneIdx].activationstatus);
    }
    BtMessageTransmittedB |= SYS_STS_REQ;
}

```

rimane fino a quando la schedulazione non è terminata o finché non viene richiesta un'attivazione manuale, in quest'ultimo caso si transita nello stato **EV_RUN_MANUAL_MODE**.

Alla fine di una fase di irrigazione automatica, viene spenta l'elettrovalvola e si ritorna allo stato **EV_IDLE**.

Quando l'**IrrigationMgr** si trova nello stato **EV_RUN_MANUAL_MODE** è in corso un'irrigazione manuale. Per ragioni di sicurezza, se l'irrigazione manuale non viene interrotta dall'utente attraverso l'App, viene interrotta automaticamente dal

software dopo un timeout di 5 minuti. Una volta terminato l'azionamento manuale si ritorna nello stato **EV_IDLE**.

L'azionamento delle elettrovalvole avviene tramite la API **I2cSlv_SendI2cMsg(I2cTxBuffer, HSD_MODULE_I2C_ADDRESS, 2)**, dove:

- **I2cTxBuffer** è un buffer che contiene il comando azionamento degli HSD che si trovano sulla SB140, è composto da due elementi:
 - 1) dal comando di accensione/spegnimento 0x50
 - 2) una maschera a bit che indica quale elettroval-

vola va accesa/spenta, ad esempio: inviando la maschera 0101 chiediamo di accendere le elettrovalvole 1 e 3 e spegnere le restanti (elettrovalvole 2 e 4), mentre con la maschera 0001 chiediamo di accendere la elettrovalvola 1 e spegnere le restanti (elettrovalvole 2,3 e 4);

- **HSD_MODULE_I2C_ADDRESS** indica l'indirizzo 0x01 che abbiamo impostato sul DipSwitch presente sulla SB140;
- **2** è la dimensione del buffer che vogliamo trasmettere sul bus I²C.

SystemTimeConfigMgr

Quando l'App si connette via Bluetooth al sistema Mercury, invia le informazioni relative al giorno, ora, minuti e secondi correnti e il **SystemTimeConfigMgr** si occupa di impostare l'RTCC Hardware attraverso una api, **Rtcm_SetRtccDate**, che il framework ci mette a disposizione, la quale riceve in ingresso una struttura che conterrà tutte le informazioni.

SystemTimeMgr

Imposta nella fase di **RTCC_INIT** tutte le informa-

zioni a 0 per transitare nello stato **RTCC_RUNNING** in cui comincerà a cadenzare il tempo.

Una volta effettuata la connessione BT con l'App, **SystemTimeMgr** conta il tempo a partire dall'ora corrente comunicatagli.

LcdMgr

Passiamo ora all'ultimo Manager, ossia quello che si occupa di mostrare sul display LCD della Expansion Board due informazioni, che sono il nome del sistema "IrrigationSystem" sulla prima riga, mentre sulla seconda riga, giorno, ora, minuti e secondi. Inizialmente la macchina a stati **LcdStatus** (descritta dallo schema a blocchi proposto nella Fig. 8) si trova nello stato di **LCD_INIT** accendiamo il display: utilizzando la API **I2cSiv_SendI2cMsg** inviamo il comando **LCD_BACKLIGHT_CMD** = 0x52 che ci dice che vogliamo accendere o spegnere la retroilluminazione del display e il parametro **LCD_BACKLIGHT_ON** = 0x01 per indicare che intendiamo accendere.

L'indirizzo da utilizzare in questo caso è quello impostato sul DipSwitch della EB210, **LCD_MODULE_I2C_ADDRESS** = 0x02

A questo punto **LcdStatus** si porta nello stato Clear in cui utilizzando il comando **LCD_CLEAR_CMD** = 0x53, le righe del display vengono resettate. Si transita a questo punto allo stato **LCD_WRITE_LINE1** in cui andiamo a selezionare la riga 1

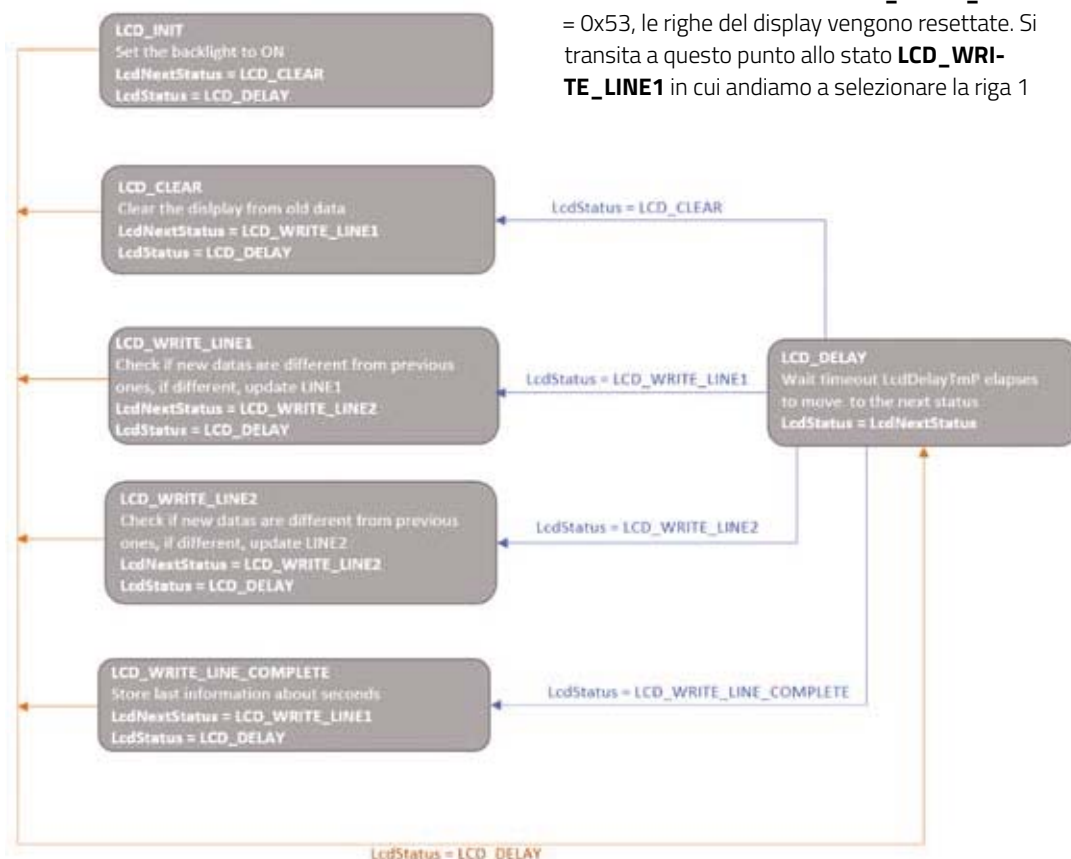


Fig. 8
Schema a blocchi della macchina a stati per LCDMgr.

CORSI FUTURA ACADEMY

Puoi scegliere tra numerosi argomenti:

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

... e molti altri ancora!



NON ASPETTARE

Se ti iscrivi in anticipo avrai diritto ad uno sconto!

INFORMAZIONI DETTAGLIATE E ISCRIZIONI SU
www.futura.academy

LCD_SELECT_LINE_1 = 0x54 per scrivere sul display "IrrigationSystem", in questo caso il comando per scrivere una linea sul display sarà **LCD_WRITE_CMD** = 0x51. Ci portiamo quindi nello stato **LCD_WRITE_LINE2** in cui andremo a selezionare la riga 2 **LCD_SELECT_LINE_2** = 0x55 e ad aggiornare l'orario se questo è diverso da quello precedentemente scritto.

CONCLUSIONI

Nei due articoli dedicati alla centralina di controllo dell'irrigazione basata su Mercury System vi abbiamo esposto come, in maniera molto semplice, siamo riusciti a scrivere il firmware della nostra Base Board BB110 senza aver dovuto gestire le configurazioni del microcontrollore o dover implementare driver specifici per i bus di comunicazione scelti. Le API messe a disposizione dal framework Mercury, ci semplificano molto lo sviluppo. Ho realizzato l'interfaccia utente, in Java con Android Studio, ma avrei potuto usare qualsiasi altro strumento, infatti, Una volta determinata la mappa messaggi, l'interfaccia utente può essere creata nel modo che preferiamo: nulla ci vieta di creare un'interfaccia web, utilizzare PyQt o un software su PC in visual basic o il sistema/linguaggio che più ci aggrada. A questo punto il nostro Irrigation System basato sul sistema Mercury è stato completamente descritto e programmato, quindi quando lo avrete terminato sarà funzionante. Concludiamo la serie di articoli riportandovi che sul nostro sito web www.elettronica.in.it insieme ai vari file dell'articolo sono disponibili per il download la mappa messaggi, i binari e i codici sorgenti del software (firmware e app) da poter utilizzare o modificare, per poter realizzare e personalizzare il vostro sistema di irrigazione. □



Cosa occorre?

Il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. La base board per il sistema Mercury (cod. BB110) è in vendita al prezzo di Euro 23,00, il modem Bluetooth (cod. MB310) costa Euro 14,00, la scheda HSD a 4 canali (cod. SB140) è disponibile a Euro 25,00, la Expansion board con LCD (cod. EB210) costa Euro 23,00 mentre la power board (cod. PB110) è in vendita a Euro 17,00. Ulteriori board sono disponibili all'indirizzo <http://bit.ly/SistemaMercury>. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



MODEM CELLULARE USB

di DAVIDE SCULLINO

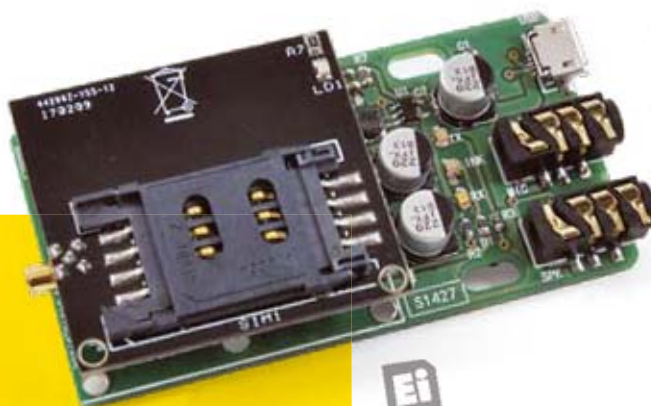


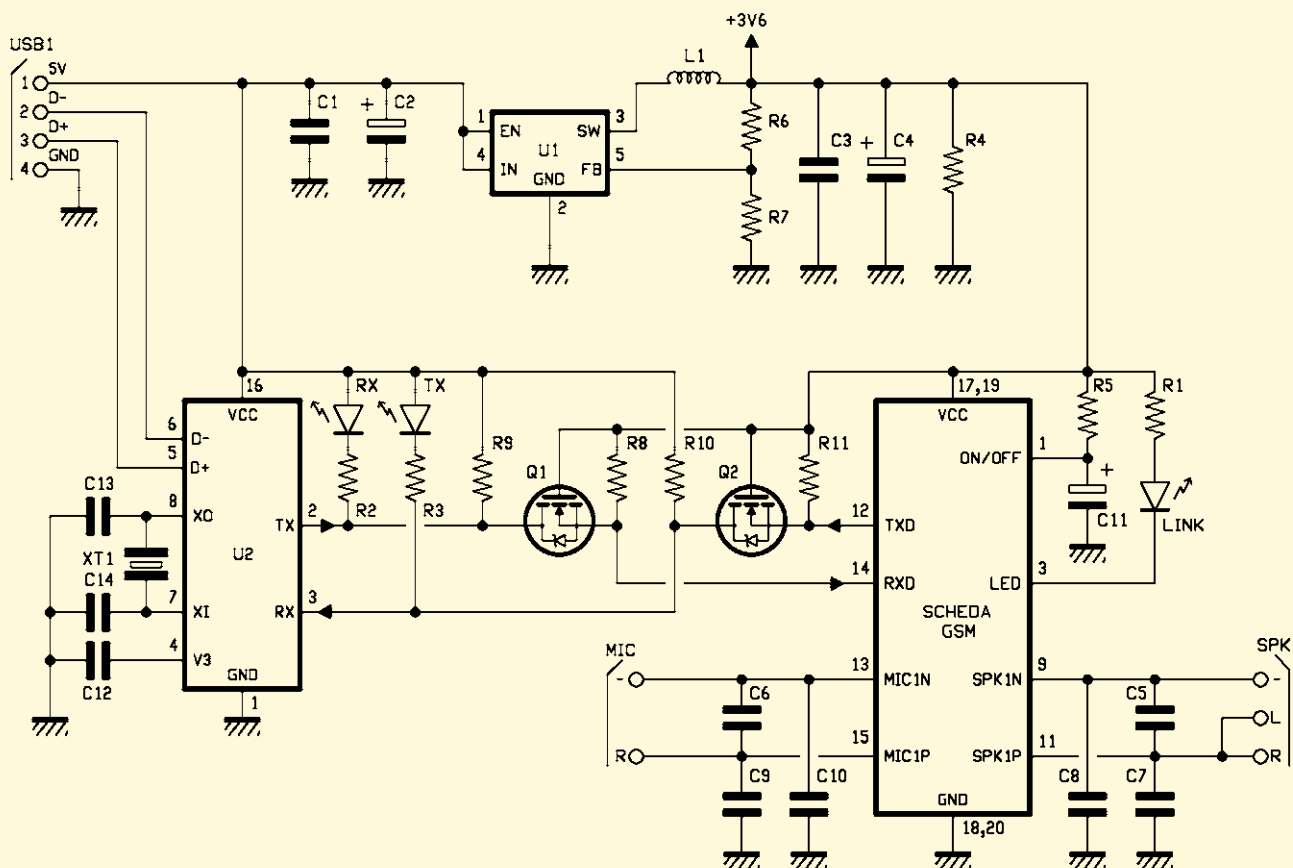
Realizziamo un dispositivo che consente la connettività GPRS tramite la porta USB di qualsiasi PC o dispositivo host USB.

L

a connettività Internet tramite rete telefonica cellulare è ormai una realtà estremamente diffusa e molti di noi ne fruiscono in modo trasparente e inconscio utilizzando oggetti di uso comune come tablet e smartphone.

Tuttavia non un fatto così scontato perché in alcuni dispositivi embedded o anche nei Personal Computer utilizzati in particolari situazioni, si può ottenere solo ricorrendo al classico modem esterno; per esempio quando non si dispone di un'infrastruttura WiFi o dell'accesso cablato a Internet, ovvero quando occorra stabilire una





connessione punto-punto per software dedicati che per indisponibilità del supporto o riservatezza non fanno passare i loro dati dal web e dai suoi server e servizi correlati. Per questa ragione abbiamo realizzato il circuito che trovate in queste pagine, che sostanzialmente è un completo modem GPRS implementato grazie all'utilizzo di un modulo cellulare SimCom SIM800C montato su un'apposita basetta che ne semplifica l'integrazione e che risparmia, a chi non è attrezzato adeguatamente, di doverlo saldare;

la scheda con il modulo già montato si trova da Futura Elettronica (on-line su www.futurashop.it) con il codice FT1308M. Il modulo viene interfacciato con un converter USB in grado di consentirne la gestione e la comunicazione attraverso una qualsiasi porta USB di un computer, di un tablet o di un apparato che disponga dell'interfaccia USB di tipo host, ivi compresi parecchi microcontrollori (Microchip, per esempio) che la integrano. Come tutti i modem GSM/GPRS, consente sia l'accesso a Internet mediante la pachetizzazione dei dati da e verso un indirizzo del web, sia l'instaurazione diretta di una comunicazione tra esso e un altro modem, di rete fissa o cellulare che sia, allo scopo di scambiarsi dati che non possono o non devono passare attraverso il web; chiaro che in quest'ultimo caso la connessione non è particolarmente veloce, avvenendo di fatto in modalità dati GSM e potendo quindi spingersi fino a un massimo di 19.200 bps. La comunicazione diretta ha quindi senso solo quando il limitato data-rate è sufficiente: per esempio per ricevere o trasmettere i record di un localizzatore GPS con connettività GSM, inviare dati di sensori remoti ecc. che trasmettono periodicamente, scambiarsi

Fig. 1
Il modulo
SIM800C.



informazioni su file riservati e comunque dove bastino i 19.200 bps consentiti. Inoltre, non meno importante è la possibilità di utilizzare il nostro modem per realizzare, insieme a un PC, una SMS machine, quindi abbinarlo a un software dedicato che permetta, tramite comandi AT, di inviare ad un database di numeri telefonici una serie di messaggi (non dimenticando i limiti imposti dal GDPR). L'utilizzo per l'invio di SMS da Personal Computer è, insieme alla possibilità di stabilire connessioni punto-punto, ciò che rende il nostro modem non sostituibile da una pur prestante chiavetta 4G.

SCHEMA ELETTRICO

Diamo dunque uno sguardo a questo piccolo e compatto modem, che è sostanzialmente composto dal modulo SimCom SIM800C montato sull'apposita scheda adattatrice (l'insieme lo trovate siglato SCHEDA GSM nello schema elettrico della pagina accanto), dal convertitore TTL/USB siglato U2 e da un regolatore di tensione; tra modulo e convertitore TTL/USB si trova il classico adattatore di livelli logici, utile perché U2 lavora a 5 volt mentre la porta seriale del modulo opera con livelli logici 0/3,3V. La scheda adattatrice (**Fig. 2**) contiene, oltre al modulo, il socket per la SIM e il connettore MMCX cui collegare l'antenna, eventualmente tramite un cavo adattatore MMCX/SMA. Dalla configurazione appare evidente che il modulo SimCom viene interfacciato all'USB mediante la porta seriale UART di cui è dotato, facente capo ai segnali TX ed RX, attraverso cui riceve i comandi di controllo (i comandi AT normali più gli estesi specifici per il modem integrato) e tramite la quale passano i dati in trasmissione e ricezione. Il SIM800C è un modulo GSM/GPRS-edge tri-band (900, 1.800, 1.900 MHz) GPRS multi-slot classe 10 (default) o 12 compatibile con lo standard GSM fase 2/2+ (2G/2,5G) capace quindi di comunicare con un data-rate teorico fino a 115 kbps (tipicamente la SimCom garantisce qualcosa di meno, ossia 85,6 kbps simmetrica, cioè sia in downlink che in uplink); può supportare gli SMS e la comunicazione vocale, a patto di connettergli un microfono e un altoparlante ai piedini preposti, che sono MIC1IN e MIC2IN (ingresso differenziale) per il microfono (da connettere al jack siglato MIC nello schema elettrico) e SPK (-, L, R) per l'altoparlante (si tratta di un'uscita a ponte, quindi sollevata da massa e pertanto da non portarne il negativo a massa). La sua gestione avviene tramite comandi AT estesi



Fig. 2
Il modulo SIM800C
montato sulla
scheda adattatrice
FT1308M.

standard GSM (set GSM07.07, 07.05) e proprietari SimCom (gli "extended AT SIMCOM command") inviati all'UART.

Il modulo è internamente coordinato da un microprocessore che gestisce la comunicazione telefonica, quella dati (mediante uno stack TCP/IP integrato) e il dialogo (tramite un UART e un'interfaccia seriale TTL) con il circuito cui il cellulare si interfaccia, ovvero con il Personal Computer. Il processore si occupa anche della gestione di una SIM da 3 o 1,8 V da montare all'esterno del modulo. Oltre a quanto detto, il SIM800C integra un'interfaccia analogica, un A/D converter, un RTC, un bus SPI, uno I²C, ed un modulo PWM. La sezione radiomobile è compatibile GSM fase 2/2+ ed il suo trasmettitore come potenza irradiata rientra in classe 4 (2 watt) a 850/ 900 MHz (GSM 850 e EGSM 900) e in classe 1 (1 watt) a 1800/1900 MHz (DCS 1800 e PCS 1900).

La parte modem supporta il PBCCH, lo stack PPP, la modalità CSD fino a 14,4 kbps e gli schemi di codifica CS 1, 2, 3, 4.

L'alimentazione prevista per il modulo è a tensione continua di 3,9 V riferita a massa e l'assorbimento alla massima potenza in trasmissione raggiunge 1 ampere.

L'interfaccia seriale TTL serve sia a comunicare i dati relativi agli SMS ricevuti che quelli in arrivo durante le sessioni TCP/IP in GPRS (il data-rate è quello previsto dal GPRS classe 10: max. 85,6 kbps), sia a ricevere i comandi dal circuito (nel nostro caso, dal PIC che governa il telecomando) che possono essere sia gli AT standard, sia quelli del set proprietario SimCom enhanced AT.

Il modulo si alimenta a tensione continua di valore compreso fra 3,4 e 4,5 V ed assorbe un massimo di 0,8 A in trasmissione; l'alimentazione viene applicata ai contatti 17 e 19 (Vcc) rispetto alla massa,

CARATTERISTICHE TECNICHE

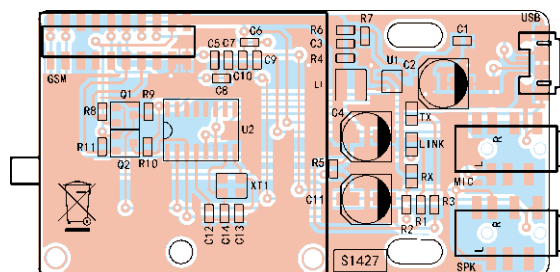
Tensione di alimentazione:
5V da USB

Corrente assorbita:
1 A

Bande di frequenza:
850/900 e 1.800/1900 MHz

Connessione dati:
GSM punto-punto e GPRS

Modalità supportate:
Modem/fax, SMS, voce



Elenco Componenti:

R1, R2, R3: 470 ohm (0603)

R4 -

R5: 56 kohm (0603)

R6: 560 kohm (0603)

R7: 100 kohm (0603)

R8, R9, R10, R11: 10 kohm (0603)

C1, C3, C12: 1 μ F ceramico (0603)

C2, C4, C11: 220 μ F 6,3 VL elettrolitico ($\varnothing$ 6mm)

C5, C6, C7, C8, C9, C10: 47 pF ceramico (0603)

C13, C14: -

XT1: -

Q1, Q2: BSS138W-7-F

MIC, SPK: Connettore Jack 3.5 da CS

USB1: Connettore micro-USB

GSM: Modulo GSM SIM800C (FT1308M)

TX: LED rosso (0805)

RX: LED giallo (0805)

LINK: LED verde (0805)

L1: Bobina 2,2 μ H

U1: LC3406

U2: CH340C

Varie

- Connettore 2x10 vie passo 2mm

facente capo ai contatti 18 e 20.

L'ingresso per il microfono MIC è filtrato da condensatori riferiti a massa e tra i due pin di input in modo da limitare la banda e trattenere eventuali disturbi captati dai collegamenti; l'uscita per l'altoparlante è anch'essa filtrata da condensatori in configurazione similare, necessari, stavolta, a sopprimere impulsi spurii derivanti dalla conversione digitale/analogica del segnale audio.

Il piedino ON/OFF del modulo serve ad accenderlo (1 logico) o a mandarlo in standby (livello basso) e in questo caso, non desiderando alcuna gestione dall'esterno lo colleghiamo alla rete R/C formata da R5 e C11, in modo da ottenere il soft-start, vale a dire da mantenerlo spento quando si applica l'alimentazione e lasciarlo accendere e diventare

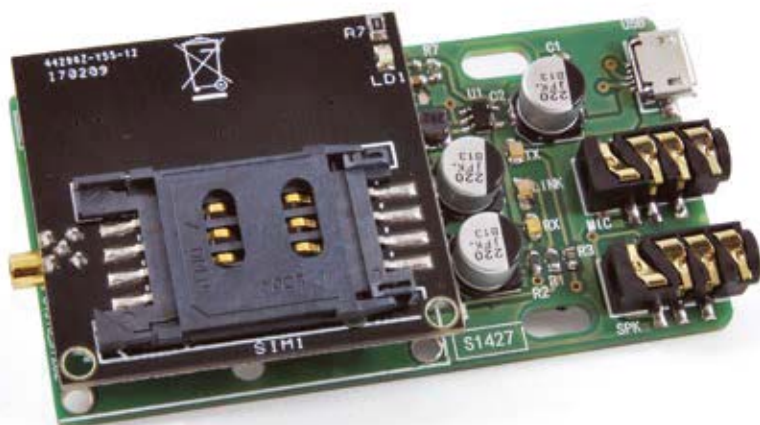
operativo dopo circa 6 secondi, che è mediamente il tempo di avvio.

L'uscita siglata LED serve a pilotare il LED di stato chiamato LINK nello schema, che di fatto corrisponde al LED di campo del cellulare, con il quale vengono segnalati la presenza della rete GSM, lo stato di connessione del modulo (assenza segnale di rete, presenza rete ecc.).

Passiamo adesso all'UART, il quale fa capo alle linee TXD (uscita) ed RXD (ingresso) interfacciate al convertitore USB/TTL mediante i traslatori di livello realizzati con Q1 e Q2, dei quali spieghiamo il funzionamento a partire dalla linea RXD e quindi dal traslatore facente capo al MOSFET Q1: quando il piedino 2 del converter U2 è a livello logico alto, il diodo di protezione interno al MOSFET Q1 (posto tra drain e source) è interdetto perché il source si trova negativo (è a 3,6 volt) rispetto al drain (che è mantenuto a 5 volt dal resistore di pull-up R9) e malgrado l'impulso non giunge all'RXD del modulo, quest'ultimo è mantenuto ad 1 logico dal proprio resistore di pull-up R8, alimentato con gli stessi 3,6V dei piedini Vcc. Invece quando il contatto TX del converter USB passa a zero logico (0 V), il MOSFET è ancora interdetto (perché il suo gate è polarizzato con 3,6 volt) ma conduce il suo diodo di protezione, dato che il source è a 3,6 V mentre il catodo è a zero, quindi il livello basso sul catodo stesso trascina a zero logico (0,6V per l'esattezza)



La scheda base del modem completa di modulo SIM800C.

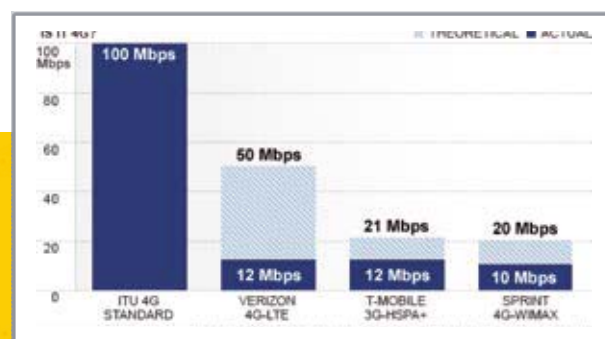


il contatto 14 del modulo GSM. Il canale RXD è unidirezionale, quindi la situazione opposta non si presenta e pertanto non è il caso di analizzarla. Passiamo adesso alla linea TXD, unidirezionale anch'essa, spiegando che quando il contatto 12 del modulo GSM è a uno logico, il diodo di protezione del Q2 conduce e riporta tale livello all'RX dell'U2; il MOSFET rimane interdetto perché source e gate sono equipotenziali. Invece a livello logico basso sul TXD, il MOSFET va in conduzione perché il suo source ora è negativo rispetto al gate, mantenuto a +3,6V e il rispettivo drain trascina a massa la linea RX dell'U2, che pertanto ripete lo zero logico. Restiamo ora un momento sul converter CH340 per analizzarne la porzione circuitale: si tratta di un device USB che realizza la funzione di convertitore bidirezionale da TTL a USB alimentato dal lato USB mediante i 5 volt dell'host giunti tramite il connettore USB1. Il componente supporta lo standard USB 2.0 ed ha un'interfaccia seriale UART full-duplex, buffer di ricezione e trasmissione integrato e supporta data-rate da 50 bps a 2 Mbps. Inoltre gestisce lato UART (anche se noi nel nostro progetto ne facciamo a meno) i segnali di controllo RTS, DTR, DCD, RI, DSR e CTS dello standard RS232. Il clock necessario al funzionamento dell'integrato viene ricavato dall'oscillatore interno, mentre il C12 è utilizzato per filtrare la tensione del regolatore interno. Su ciascuna delle linee TX ed RX è applicato un LED di stato polarizzato attraverso un resistore, il tutto collegato al positivo dei 5V: ciascun LED segnala l'attività dei canali dati di ricezione e trasmissione sull'interfaccia UART. Concludiamo l'analisi circuitale con il regolatore di

tensione U1, qui utilizzato per ricavare l'alimentazione del modulo GSM a partire dai 5 volt disponibili sull'USB; il componente è un LC3406 e si tratta di un DC/DC step-down ad alta efficienza (96%) di tipo buck, capace di erogare una corrente di ben 1,2 A (necessaria, visto che il modulo alla massima potenza in trasmissione assorbe circa lo stesso). L'integrato accetta tensioni continue d'ingresso nel range tra 2,6V e 7V e fornisce una tensione d'uscita definita dal partitore resistivo R6/R7, il quale riporta una porzione della tensione d'uscita (prelevata a valle dell'induttore di filtro L1, pilotato dall'uscita SW) all'ingresso di retroazione FB; il campo della tensione d'uscita può variare da 0,6V e il valore applicato tra il piedino IN e massa. In virtù dell'elevata frequenza di commutazione (ben 1,5MHz) i componenti di filtro L1 e C3-C4 possono avere valori e quindi dimensioni molto contenuti, a tutto vantaggio della miniaturizzazione del circuito. L'integrato LC3406 contempla una modalità di standby attivabile mediante il piedino EN, che permette di lasciare il componente alimentato, disattivandolo o rendendolo operativo mediante applicazione di un livello logico ad esso; per l'esattezza, ponendo EN al potenziale di ingresso U1 si attiva, mentre va in standby, assorbendo una corrente trascurabile (inferiore al microampere), quando EN si trova a zero volt, ossia al potenziale del piedino GND o comunque a meno di 0,6V. L'integrato LC3406 è incapsulato in un package SOT23-5 per montaggio superficiale.

PIÙ VELOCI COL WiFi O SU RETE MOBILE?

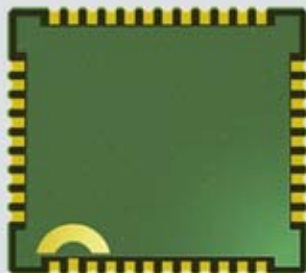
Per quanto la rete telefonica cellulare sia ancora eterogenea, secondo una ricerca sembra che nei 33 Paesi in cui è stato eseguito il test, l'accesso a Internet tramite WiFi e rete cablata garantisca mediamente meno velocità di quella permessa dalla connessione dati degli smartphone. La ragione è più d'una, ma a parte le limitazioni degli hot-spot, che devono dividere la loro banda tra tutti gli utenti collegati, è da ricondurre alle infrastrutture cablate, che non



sempre garantiscono velocità di connessione sufficienti; questo accade nei Paesi dove Internet viene ancora servito attraverso linee telefoniche tradizionali per l'ultimo miglio e nelle nazioni in cui la rete cablata è vecchia o non è stata sviluppata, perché più costosa da realizzare di una mobile, che richiede meno lavoro e manutenzione. Quest'ultimo è il caso dei Paesi in via di sviluppo, dove la telefonia è stata realizzata negli anni di quella mobile.

Il modulo che realizza la connessione GSM/GPRS e quindi il cuore del modem è il SIM800C della SIMCom: si tratta di uno dei più recenti moduli cellulare Quad-band compatibili GSM/GPRS, operante sulle frequenze di 850/900/1800/1900MHz, utilizzabile sia per accessi a Internet, sia per comunicazione vocale (a patto di collegargli un microfono e un piccolo altoparlante) che SMS. Esternamente si presenta in un package grande 24mm x 24mm x 3 mm con contatti posti sui quattro lati e sagomati ad L in modo da poter essere saldati di lato e sotto. Il modulo è internamente gestito da un processore AMR926EJ-S che gestisce la comunicazione telefonica, quella dati (mediante uno stack TCP/IP integrato) e il dialogo (tramite un UART e un'interfaccia seriale TTL) con il circuito cui il cellulare si interfaccia. Il processore si occupa anche della

gestione di una SIM da 3 o 1,8 V da montare all'esterno del modulo. Oltre a quanto detto, il SIM800C integra un'interfaccia analogica, un A/D converter, un RTC, un bus SPI, uno I²C, ed un modulo PWM. La sezione radiomobile è compatibile GSM fase 2/2+ ed è di classe 4 (2 W) a 850/ 900 MHz e di classe 1 (1 W) a 1800/1900 MHz. L'interfaccia UART (seriale TTL) serve sia a comunicare i dati relativi agli SMS ricevuti che quelli in arrivo durante le sessioni TCP/IP in GPRS (il data-rate è quello previsto dal GPRS classe 10: max. 85,6 kbps), sia a ricevere i comandi dal circuito (nel nostro caso, dal PIC che governa il telecomando) che possono essere sia gli AT standard, sia quelli del set proprietario SIMCom enhanced AT. Il modulo si alimenta a tensione continua di valore compreso fra 3,4 e 4,5 V ed assorbe un massimo di 0,8 A in trasmissione.



REALIZZAZIONE PRATICA

Per quel che riguarda la realizzazione, il modem richiede una certa attenzione, in quanto è composto principalmente da componentistica a montaggio superficiale; comunque per semplificarvi il compito il modulo SIM800C è reperibile presso Futura Elettronica (www.futurashop.it) già montato sull'apposita scheda adattatrice che dal lato opposto riporta il socket per la SIM-Card e che integra il connettore a doppia fila 2x10 poli a passo 2,54 mm. Una volta procurato il modulo, la prima cosa da fare è preparare il circuito stampato a doppia faccia del quale trovate le tracce del circuito stampato sul nostro sito web, www.elettronica.it, che userete per preparare le pellicole e procedere con la fotoinci-

sione. Tutte le saldature dovranno essere effettuate con un saldatore da 20÷25 W di potenza a punta molto sottile, usando filo di lega saldante da 0,5 mm di diametro; meglio sarà disporre di una lente d'ingrandimento e una pinzetta per posizionare i componenti, nonché di flussante in pasta da deporre sulle piazzole dei componenti prima di saldare, così da agevolare lo scioglimento e il posizionamento dello stagno. Una volta ottenuto il circuito stampato, il primo componente da montare è il converter CH340, centrandolo bene nelle rispettive piazzole e stagnando un pin alla volta; passate poi al connettore femmina a 2 file da 10 poli a passo 2,54 mm che suggeriamo di posizionare ben centrato sulle piazzole corrispondenti e di fermare stagnando leggermente un elettrodo per lato; nel fare le saldature, raccomandiamo di prestare molta attenzione a non cortocircuitare, con lo stagno, piedini vicini. Utilizzate la stessa tecnica per il regolatore di tensione U1 e poi dispo- nete gli altri componenti, in ordine di altezza. Fate attenzione ai LED, che dovrete cercare di saldarli scaldandoli il meno possibile e ricordando, a circuito ultimato, di pulire le eventuali tracce di flussante che dovessero rimanervi sopra; altrimenti la luce che emetteranno verrà ridotta e deviata. Passate poi al montaggio dei connettori jack, di tipo SMD anch'essi, ed il microUSB (ovviamente SMD). Completate le saldature dei componenti più critici e accertato che non vi sono piazzole unite accidentalmente, si può passare a tutti gli altri componenti, iniziando con quelli a più basso profilo e rispettando le polarità indicate dai disegni di montaggio illustrati in queste pagine. Una volta terminato, il modem può essere inserito in un contenitore in plastica tipo quello che vedete nelle foto del prototipo in queste pagine: si tratta di un box in plastica delle dimensioni di 20x40x73 acquistabile dalla Futura Elettronica con il codice prodotto 5110-SC-703, che dovrete lavorare come visibile nelle fotografie, ossia ricavando la finestrella per il connettore microUSB più i fori per i jack su una delle facce piccole e il foro per il connettore dorato della scheda GSM sulla faccia opposta. La **Fig. 3** fornisce un'idea di come montare il circuito nel contenitore suddetto e in che modo lavorare il contenitore per ricavare le aperture del caso; tra queste non figura quella per la SIM-Card in quanto la stessa si inserisce nel lettore della scheda GSM prima di chiudere il contenitore. Per l'alimentazione utilizzerete lo stesso connettore microUSB, che oltre che per la comunicazione con il computer o altro dispositivo host, servirà per

fornire l'elettricità che serve.

Per l'utilizzo del modem con i Personal Computer dovete scaricare da Internet i driver del convertitore USB CH340 (per esempio da <https://www.arduino.eu/ch340-windows-8-driver-download/> o dalla pagina web <https://www.futurashop.it/chip-converter-USB-SERIALE-3085-CH3085>), dove li trovate per i sistemi operativi Windows, Linux e MacOS. Una volta scaricati i driver e collegato il modem tramite un cavo USB adatto, quando verrà indicato che è stato trovato nuovo hardware procedete selezionando il percorso di ricerca e installando i driver. Questa è la procedura per Windows, mentre in Linux e MacOS le cose possono differire. Ricordate che come tutti i modem cellulari, anche il nostro richiede che il modulo GSM/GPRS sia dotato di una SIM attiva e, se ricaricabile, contenente credito a sufficienza; la SIM-Card deve inoltre essere abilitata al traffico dati, cosa che di norma nelle SIM moderne è implementata senza bisogno di interventi esterni, richieste di attivazione ecc.

CONCLUSIONI

Il modem qui descritto rappresenta una soluzione pratica e portatile per dotare di connettività GSM e Internet, sia pure di velocità che oggi può essere considerata limitata (non dimentichiamoci che siamo nell'era dell'LTE, dove data-rate di decine di Mbps sono facilmente ottenibili), Personal Computer utilizzati sul campo o dove non vi sia la rete cablata per l'accesso a Internet o in mobilità, ma in generale per tutti quei dispositivi che non integrano una connessione cellulare nativamente, come gli smartphone e i moderni tablet. La praticità di poterlo alimentare tramite la stessa



Fig. 4

Dal lato della scheda GSM occorre un foro per affacciare all'esterno il connettore d'antenna.

Fig. 3

La scheda base del modem completa di modulo SIM800C.



porta USB utilizzata per la connessione dati e le ridottissime dimensioni (superiori a quelle di una "chiavetta modem ma pur sempre contenute) e il peso contenuto, lo rendono un utile accessorio da portare con sé negli interventi sul campo o nelle località sprovviste di connessione cablata.

Il modem non nasce per competere con le più recenti chiavette 3G ed LTE ma piuttosto per applicazioni dove serva la connessione occasionale o serva un apparato GSM capace di mettersi in contatto con uno analogo (o altro modem GSM) per instaurare un collegamento punto a punto attraverso cui ritrasmettere quantità limitate di dati, cosa che con le chiavette non è fattibile perché nascono per l'accesso a Internet e il supporto al protocollo TCP/IP e non alla comunicazione diretta modem-modem. ■



Cosa occorre?

Il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. La base del modem GSM (cod. FT1427) è in vendita al prezzo di Euro 13,00, il contenitore plastico da forare (cod. SC-703) è disponibile a Euro 1,10, la breakout con SIM800C (cod. FT1308M) costa Euro 29,00, il modem con M95 (cod. FT1128M) è disponibile a Euro 39,00, la breakout con il modulo GSM/GPS (cod. FT1178M) è in vendita a Euro 59,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Sostituire il telecomando TV? Nulla di più facile!

Il tuo telecomando TV è rotto o non funziona più bene?
È arrivato il momento di sostituirlo con i telecomandi universali a infrarossi compatibili con le TV delle maggiori marche prodotte dal 2000 ad oggi. Dispongono di tutte le funzioni del telecomando originale. Subito utilizzabili!



PRONTI ALL'USO



Cod. **TVIR-SAMSUNG**



Cod. **TVIR-SONY**



Cod. **TVIR-PHILIPS**



Cod. **TVIR-LG**



Cod. **TVIR-PANASONIC**

€ 5,90
OGNI PEZZO

DA PROGRAMMARE



€ 9,00
cod. **TVFREEDOM2**

2 in 1

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

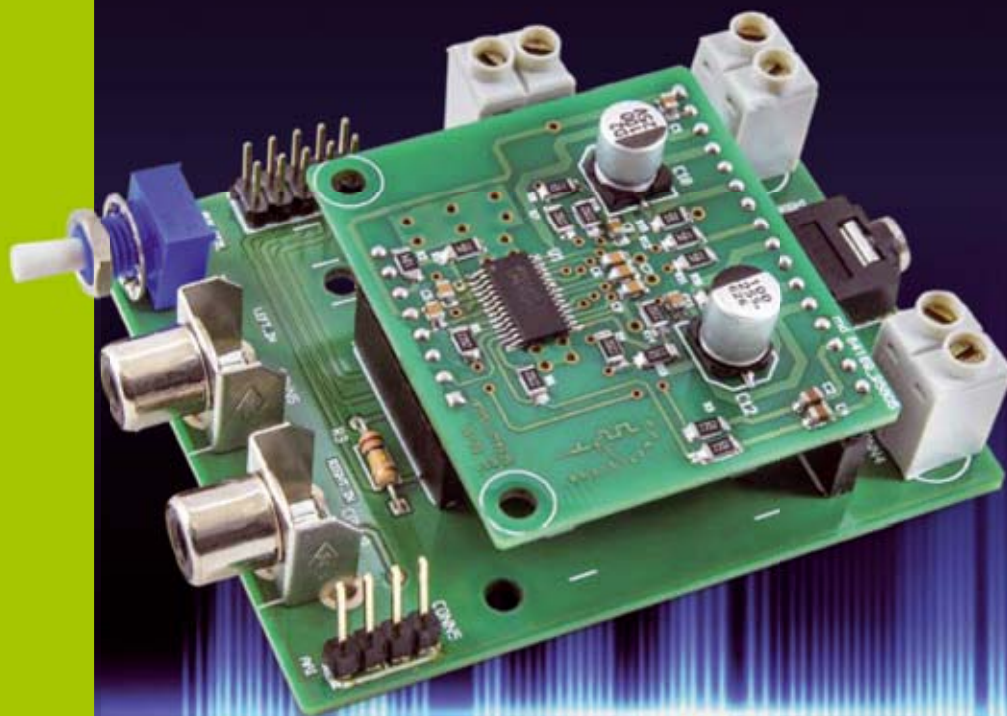
€ 8,50

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



AMPLIFICATORE STEREO UNIVERSALE

di LAPO PIERI

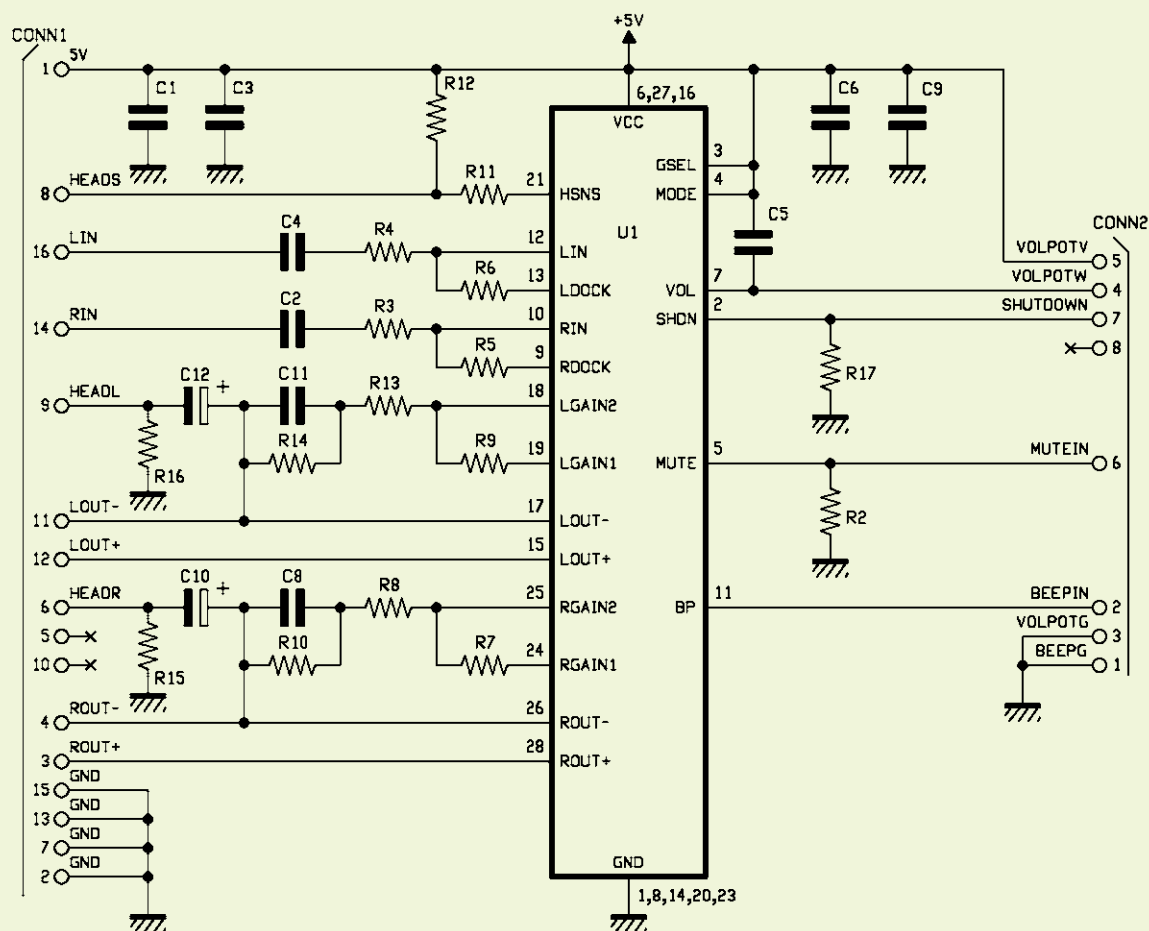


Composto da una scheda base su cui montare un modulo finale a scelta tra LM4835 o PAM8403, permette di realizzare amplificatori audio per ogni applicazione, sia per cuffie che per pilotare una coppia di altoparlanti.

N

on sempre nell'audio domestico o professionale servono i "watt forti" perché spesso ci bastano piccole potenze per l'ascolto in cuffia o in piccoli monitor da tenere sul banco o sulla scrivania; in casi come questi gli amplificatori BF preferiti sono quelli basati su un singolo

chip, che un tempo era il classico contenente soluzioni circuitali analogiche in classe AB (tipici sono gli LM386N, TDA2030, TDA2004 ecc.) e che oggi ha spesso come cuore un elemento in classe D. Si tratta sempre di finali analogici, giacché nella classe D, pur ricostruendo il segnale di uscita con impulsi



rettangolari, la loro larghezza varia in analogia all'ampiezza del segnale da amplificare, però rispetto agli amplificatori vecchio stampo consuma meno elettricità, quindi scalda meno e non richiede, almeno nelle piccole e medie potenze (anche decine di watt per canale), alcun dissipatore di calore. In queste pagine vogliamo sottoporvi il progetto di un amplificatore stereo per utilizzo generico che grazie alla realizzazione modulare ci consente di scegliere tra due potenze d'uscita e due classi di funzionamento, così da adattarsi ad ogni esigenza; la modularità è garantita dal fatto che il sistema è composto da una motherboard contenente le connessioni con l'esterno (ingresso e uscita, alimentazione, segnali di controllo) sulla quale, mediante appositi connettori, possiamo montare un modulo basato su un nuovissimo integrato della Texas Instruments da 2x2W oppure un altro, in classe D, realizzato con il PAM8403, già utilizzato in nostri progetti e capace di sviluppare 2x3 watt. Del PAM 8403 abbiamo spiegato diffusamente in vari progetti del passato, quindi concentreremo la

nostra attenzione sull'LM4835, che è un piccolo amplificatore audio stereo in classe AB provvisto anche di un'uscita per le cuffie; possiede vari ingressi di controllo pur mantenendo uno schema semplice e il numero di componenti esterni, necessari al funzionamento, particolarmente limitato.

IL PROGETTO

L'amplificatore si basa su una demoboard, che possiamo considerare una scheda madre (non a caso chiamata motherboard) sulla quale poter montare il modulo contenente il solo LM4835 più i pochi componenti passivi necessari al suo funzionamento (che formano un circuito stampato a se stante) e in alternativa il modulo già presentato sulle pagine di Elettronica In e reperibile da Futura Elettronica con il codice 2846-MICROAMP-LID2X3W e basato sul PAM8403. Quest'ultimo modulo -a fronte di un costo minore- ha anche una minore versatilità e non possiede l'uscita separata per le cuffie. Ma concentriamoci sull'LM4835 e sul modulo che

lo ospita, il cui schema è illustrato nella pagina accanto: questo integrato, prodotto dalla Texas Instruments, contiene tutti gli stadi necessari a realizzare un completo amplificatore in classe AB. Il relativo modulo può sia essere utilizzato da solo o integrato in altri circuiti, sia montato sulla motherboard.

Il nostro integrato riesce a fare tutto in un package TSSOP a 28 pin (esiste anche in versione WQFN, ma noi abbiamo optato per il TSSOP...) perché racchiude tutti gli stadi necessari, vale a dire un preamplificatore stereo d'ingresso, un blocco di controllo del volume a 32 passi comandato in tensione, un gestore dell'alimentazione e una logica che governa il MUTE e lo shutdown.

Partiamo dalle uscite del finale, che sono a ponte utilizzabili anche in modalità single-ended: la differenza è che nel primo caso l'altoparlante di ciascun canale si collega tra il + e il - della relativa uscita, mentre nel secondo tra il + e la massa comune. In altre parole, come visibile nello schema elettrico del modulo, l'uscita sinistra fa capo ai punti LOUT- e LOUT+ del connettore CONN1 mentre quella destra è localizzata tra ROUT+ ed ROUT-; con tale

configurazione il nostro amplificatore eroga un massimo di 2W su 4 ohm, che si riducono a circa 1 su 8 ohm, ovvero a 4 ohm ma collegando l'altoparlante in single-ended, ovvero tra il + e la massa. Nel caso dell'uscita destra, tra ROUT+ e GND e per la sinistra, fra LOUT+ e GND; notate che in modalità single-ended l'altoparlante di ciascuna uscita dev'essere disaccoppiato in continua mediante un condensatore elettrolitico (tipicamente da 470 μ F 6,3V o anche 1.000 μ F 6,3V) mentre a ponte ciò non è richiesto, giacché i piedini LOUT+/-, come gli ROUT+/-, pur presentando una tensione continua sono equipotenziali e quindi idealmente nell'altoparlante, a riposo, non fluisce alcuna corrente. Notate che in teoria il rapporto tra la potenza erogata a ponte e quella in single-ended sarebbe circa di quattro a uno, tuttavia in questo caso la potenza a ponte è limitata internamente.

Questo è quanto riguarda le uscite per gli altoparlanti, ma nel nostro circuito abbiamo previsto anche le uscite per cuffia, che sono a più basso livello, dovendo pilotare trasduttori da pochissime centinaia di milliwatt e impedenza tipicamente di 32 ohm; tali output fanno capo ai contatti HEADR

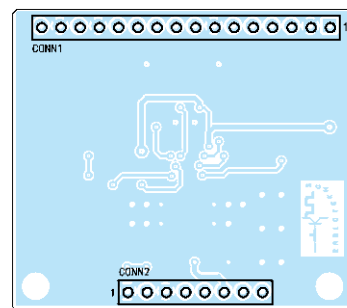
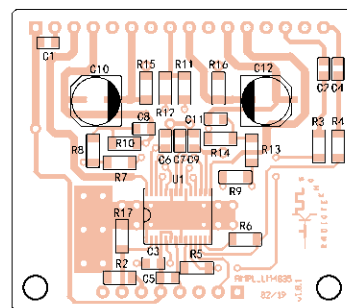
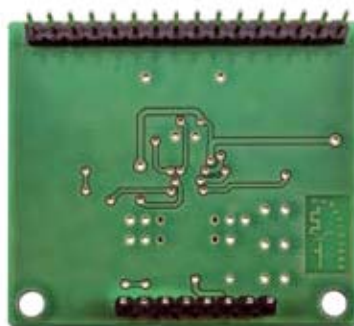
piano di montaggio LM4835

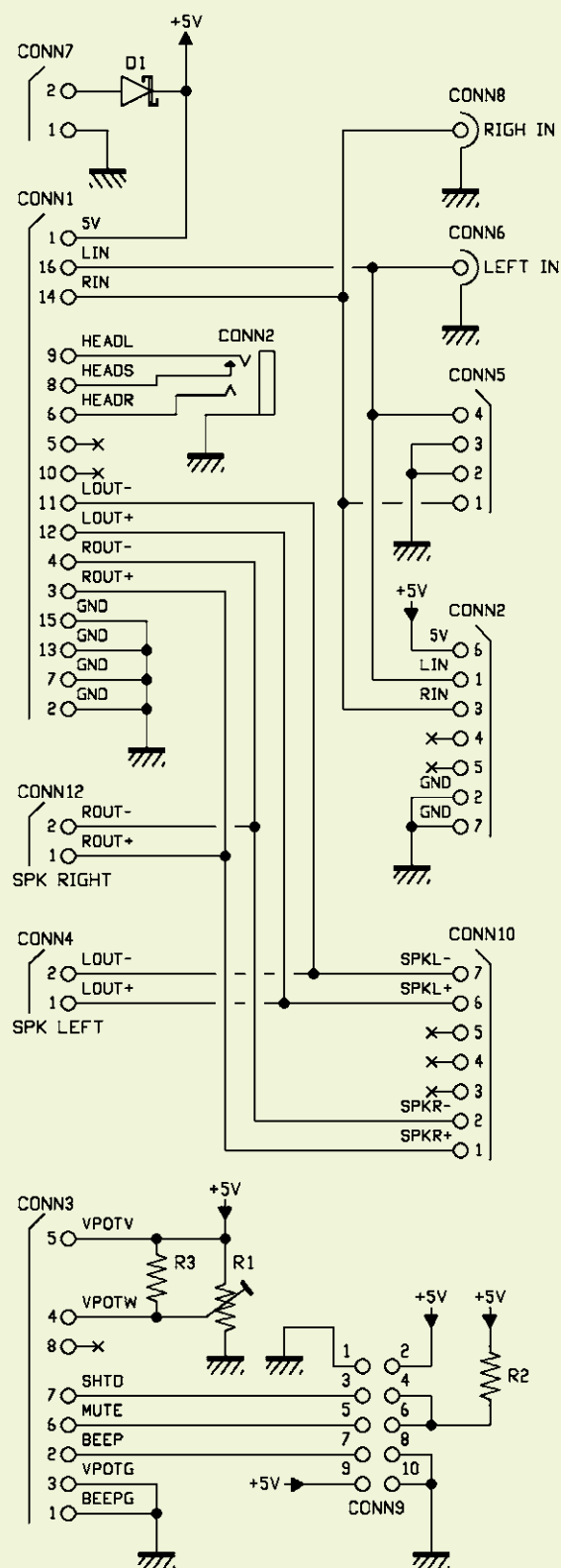
Elenco Componenti:

R10, R13, R14, R3, R4,
R5, R6, R7, R8, R9: 22 kohm (1206)
R15, R16: 1 kohm (1206)
R11, R12, R17, R2: 100 kohm (1206)
C1: 4,7 μ F (0805)
C2, C4: 470 nF ceramico (0805)
C3, C5, C6, C7, C9: 100 nF ceramico (0805)
C10, C12: 100 μ F 16 VL elettrolitico ($\varnothing$ 6mm)
C11, C8: 68 nF ceramico (0805)
U1: LM4835MTX/NOPB
CONN1: Strip maschio 16 vie
CONN2: Strip maschio 8 vie

Varie:

- Circuito stampato S1441 (45x40 mm)





(per il canale destro) e HEADL (per il sinistro) e sono riferite a GND.

Le uscite per cuffia sono disaccoppiate mediante i condensatori elettrolitici C12 e C15, perché sono riferite a massa e pertanto vale quanto spiegato per la modalità single-ended a proposito del collegamento degli altoparlanti.

Le resistenze R15 ed R16 servono a scaricare i suddetti elettrolitici quando si spegne l'amplificatore e in assenza di segnale, onde evitare di sentire il "botto" in cuffia quando si connette la stessa inserendone lo spinotto nella presa jack della scheda madre.

A proposito di cuffia, il comune dell'uscita corrispondente è la massa, che vedrete sulla scheda madre. Il pin 8 del connettore CONN1 fa capo al piedino HSNS, che normalmente va chiuso su HEADL e che se viene aperto dall'inserzione dello spinotto nel jack della scheda madre fa spegnere le uscite degli altoparlanti. Come vedrete analizzando lo schema elettrico della motherboard, tale pin viene commutato dall'interruttore della presa per cuffia.

La retroazione degli amplificatori d'uscita è realizzata con le reti C8/R10, R7, R8 per il canale destro e C11/R14, R9, R13 per il sinistro.

Quanto agli ingressi, il circuito li ha, sempre sul connettore CONN1, localizzati sui contatti LIN ed RIN, rispettivamente per il canale sinistro e il destro: ciascuno viene disaccoppiato in continua da un condensatore (C4 per LIN e C2 per RIN) e il rispettivo stadio preamplificatore, che precede il blocco interno di controllo del volume, retroazionato mediante R6 ed R4 per il canale sinistro ed R5/R3 per il destro. Il guadagno degli stadi di ingresso è quindi determinato dal rapporto tra i due resistori, ovvero $R6/R4$ o $R5/R3$.

Per il controllo del volume è stato previsto un ingresso a livello di tensione sul connettore CONN2, facente capo al contatto VOLPOTW; nel nostro caso, sulla motherboard è collegato a un potenziometro con un estremo a VOLPOTV (ossia a 5V) e uno a GND ma comunque in linea generale il potenziometro dev'essere alimentato dal pin VOLPOTV.

Concludiamo l'analisi del modulo con gli ingressi di MUTE e shutdown, entrambi attivi a livello alto e mantenuti normalmente a zero (quindi disattivati) dalle resistenze R2 e R17; per livello 1 si intende il potenziale di alimentazione, ossia quello applicato ai pin Vcc (6, 16, 27).

Il comando di MUTE permette di spegnere lo stadio finale e quindi tacitare all'istante gli altopar-

lanti; si applica al contatto 6 (MUTEIN) del CONN2; invece quello di shutdown è localizzato sul pin 7 (SHUTDOWN) del CONN2 e manda l'integrato in standby, spegnendone tutti gli stadi interni ad eccezione di quello di controllo (Mode Control) cui fa capo e limitando, così, l'assorbimento ad appena 0,7 microampere!

Sulla scheda dell'LM4835 abbiamo previsto anche l'ingresso BEEPIN (CONN2) che serve per sovrapporre un tono, tipicamente composto da un'onda rettangolare, all'audio: questo risulta utile per fornire segnalazioni durante l'ascolto, come ad esempio una sveglia o un timer, ovvero un allarme, provenienti da un circuito esterno.

L'integrato LM4835 non necessita di alcun dissipatore, perché il circuito stampato, con le sue piste in rame, è sufficiente a dissipare il poco calore generato in condizioni di utilizzo normale. Tuttavia il produttore consiglia, in caso di utilizzo prolungato alla massima potenza e a temperature ambiente elevate (30 °C e più) di provvedere a una modesta circolazione d'aria forzata ottenuta tramite una piccola elettroventola.

LA SCHEDA MADRE

Passiamo adesso alla descrizione della motherboard pensata per ospitare i due moduli: quello appena descritto, basato sull'LM4835, oppure il circuito con PAM8403. Per ciascuno la scheda prevede una coppia di strip femmina a passo adeguato.

Sulla motherboard troviamo essenzialmente i connettori RCA per gli ingressi sinistro e destro, le morsettiere per connettere gli altoparlanti, la morsettiera per l'alimentazione e pochi altri componenti: vediamoli insieme.

Partiamo dagli strip per il modulo appena descritto, che sono CONN1 e CONN3: il primo corrisponde all'omonimo sul modulo e il CONN3 al CONN2 dello stesso; quanto al modulo con il PAM8403, se optate per utilizzare questo (l'uno è chiaramente alternativo all'altro) disponete dei CONN2 e CONN10. Le morsettiere CONN4 e CONN12 permettono il collegamento degli altoparlanti, che è previsto in modalità a ponte; se volete la connessione single-ended dovete connettere un solo morsetto di ciascuna al + dell'altoparlante e il – dello stesso a GND.

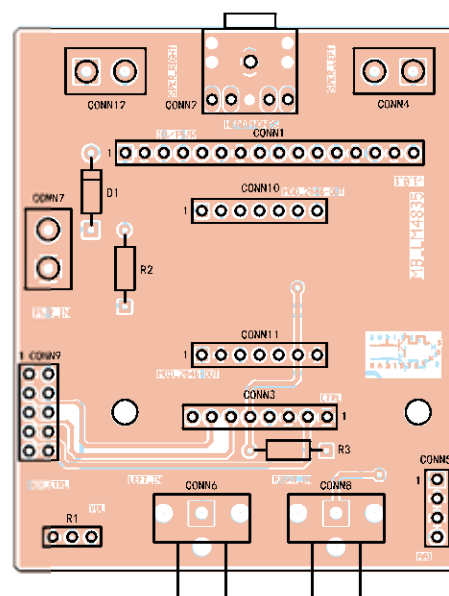
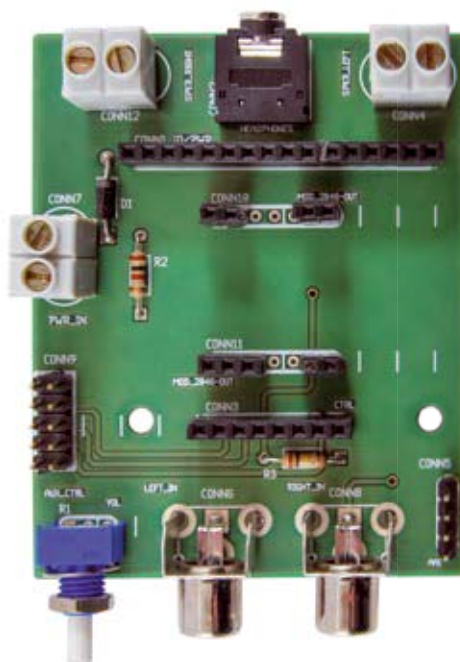
piano di montaggio MOTHERBOARD

Elenco Componenti:

- R1: 10 kohm trimmer
- R2: 100 ohm
- R3: 100 kohm
- D1: 1N5819
- CONN2: Connettore jack stereo 3.5mm da c.s.
- CONN6, CONN8: Connettore RCA Femmina da c.s.
- CONN1: Strip femmina 16 vie
- CONN3: Strip femmina 8 vie
- CONN9: Strip maschio 2x5
- CONN5: Strip maschio 4 vie

Varie:

- Circuito stampato S1440 (59x72 mm)



Il progetto descritto in queste pagine è centrato sull'integrato LM4835 della Texas Instruments, un monolitico in SMD a 28 contatti che contiene un completo amplificatore stereo con due uscite a ponte per altoparlanti e due per cuffia che una volta collegate consentono di disattivare gli altoparlanti.

All'interno del componente troviamo uno stadio preamplificatore d'ingresso con linea separata per sovrapporre un tono acustico (BEEP IN), un regolatore di volume (potenziometro elettronico) comandato in tensione e una logica di controllo che governa sia la funzione MUTE (la quale tacita gli altoparlanti e la cuffia) che lo shutdown (spegne quasi tutto l'amplificatore, riducendo l'assorbimento a 0,7 μ A).

Le uscite per altoparlante, se utilizzate a ponte possono erogare ciascuna 2W su 4 ohm (con distor-

sione THD dell'1%) o 2,2W su 3 Ω .

All'interno troviamo anche un circuito di soppressione dei rumori di Click & Pop e una Thermal Shutdown Protection che spegne lo stadio finale in caso di superamento della temperatura massima ammissibile.

Le potenze ottenibili dall'LM4835 sono per quanto riguarda le uscite a ponte per altoparlanti è:

- 1,1 W (con 1% di THD) a 8 Ω ;
- 2,2 W (con 1% di THD) su 3 Ω ;
- 2.2 W ((con 1% di THD) .

Invece l'uscita per cuffia, la quale funziona solo in modalità single-ended, eroga tipicamente 85mW per canale su impedenza di 32 Ω . La fedeltà sonora dell'integrato, che non ha nulla da invidiare agli amplificatori più potenti, è enfatizzata dalla modalità Boomer, che rinforza i toni bassi dando la sensazione che la potenza sonora sia più elevata di quella reale.

Per D1 è stato scelto uno Schottky (1N5819) allo scopo di limitare la caduta di tensione.

Gli ingressi audio sono localizzati sui connettori RCA da circuito stampato siglati, nello schema, CONN6 (sinistro) e CONN8 (destro).

Oltre a quelli descritti, abbiamo anche un connettore multifunzione siglato CONN9, dove sono riportati i pin di shutdown (per ottenerlo bisogna ponticellare i piedini 3-4), quelli del MUTE (ponticellare 5-6), quello dell'ingresso per il BEEP (che deve essere riferito a massa) oltre a GND e al +5V. Sempre sulla motherboard per l'ingresso audio è stato previsto un connettore pin-strip a quattro poli (connettore CONN5).

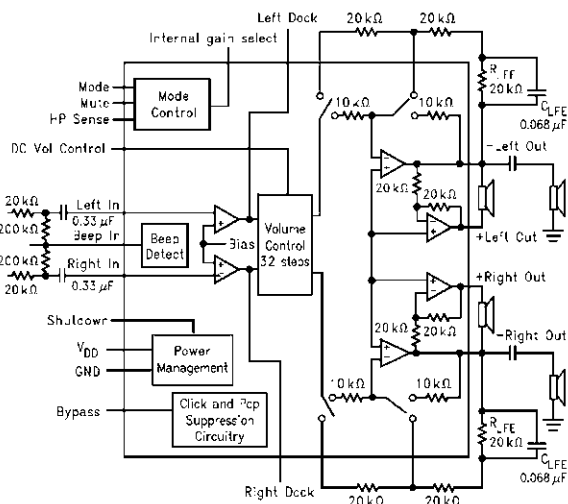
Per quanto riguarda il controllo del volume dell'LM4835, è sufficiente una tensione continua che farà variare il volume proporzionalmente da zero al massimo; tale tensione, da riferire a massa, spazierà tra circa 0 (minimo volume) e 5V (massimo volume). Per generarla, sulla motherboard abbiamo previsto il potenziometro R1 da 10 kohm, il cui cursore è applicato al piedino VPOTW del CONN3. Se pensate a un controllo dall'esterno, dovete rimuovere il potenziometro e applicare la relativa tensione tra VPOTW e massa.

REALIZZAZIONE PRATICA

Ciò detto possiamo passare all'aspetto pratico, ossia alla costruzione del modulo con LM4835 e della motherboard; per entrambi è stato previsto un circuito stampato a doppia faccia ottenibile per fotoincisione utilizzando, per creare le pellicole, le tracce lato rame scaricabili dal nostro sito Internet www.elettronica.in.it tra i file del progetto.

La realizzazione della motherboard è veramente semplice, visto che richiede componenti unicamente a foro passante (THT): connettori audio, connettori strip femmina (S.I.L. a passo 2,54 mm), nonché il potenziometro del volume, il jack da 3,5 mm stereo per la cuffia e i connettori di alimentazione, oltre a due resistenze e un diodo di protezione, utile a evitare di danneggiare i moduli in caso di inversione della polarità dell'alimentazione.

Un po' più complesso è il montaggio del modulo contenente l'LM4835, perché impiega componenti a montaggio superficiale (i componenti passivi più piccoli sono in formato 0805 e gli altri in 1206) e gli unici elementi a foro passante sono i due pin-strip che ne permetteranno l'inserzione nella motherboard. Comunque basta armarsi di pazienza e di un minimo di attrezzatura consistente in un saldatore a punta fine da 20+25W, del filo di lega saldante del diametro massimo di 0,5 mm, una pinzetta per



L'alimentazione per il circuito dev'essere di 5V stabilizzati (la corrente è dell'ordine di 0,8A per il modulo con LM4835 e poco più di 1 A se optate per il PAM8403) applicati alla morsettiera siglata CONN7, la cui linea positiva è protetta dal diodo D1 che serve a impedire il danneggiamento dei moduli in caso di inversione accidentale della polarità.

posizionare i componenti, della pasta flussante da spalmare sulle piazzole prima di saldare i componenti e una lente di ingrandimento per aiutarvi nel posizionamento e per verificare, alla fine del montaggio, se tutto è montato correttamente e non vi sono baffi di stagno a creare cortocircuiti.

L'integrato LM4835 andrà montato per primo, proseguendo poi in ordine di altezza con i componenti restanti, lasciando per ultimi gli elettrolitici. Visto il passo molto ridotto del componente, in contenitore TSSOP con piedini a passo 0,65mm, raccomandiamo di cospargere le relative piazzole di flussante prima di saldarle, così da evitare che la lega saldante unisca pin che non devono essere in contatto.

Dopo l'integrato potranno essere montati tutti i restanti componenti passivi, resistenze e condensatori, e solo alla fine i due pin-strip che lo faranno diventare un modulo pronto per essere utilizzato su di un altro circuito stampato (come la motherboard ma non necessariamente) oppure addirittura su una millefiori.

Il modulo non presenta alcuna protezione, quindi fate attenzione ai cablaggi, soprattutto se lo utilizzate senza la motherboard: il piedino 1 è indicato dalla piazzola quadrata (le altre sono circolari) oltre che dalle indicazioni sul solder-resist del circuito stampato.

La messa in funzione non richiede accortezze particolari; si consiglia, però, di partire con il volume al minimo e con un basso livello di segnale in ingresso.

Per quanto riguarda la scelta degli altoparlanti, l'LM4835 si adatta ad impedenze da 3 a 8 ohm oltre a poter pilotare le classiche cuffie da 32ohm nominali. Simile è il discorso se decidete di montare, sulla motherboard, il modulo basato sul PAM8403.

A proposito di montaggio, la **Fig. 1** mostra come posizionare sulla motherboard il modulo basato sull'LM4835, mentre in **Fig. 2** trovate la stessa motherboard con a bordo il modulo con PAM8403; per montare quest'ultimo, sebbene si acquisti già montato dovete applicare dei pin-strip in corrispondenza delle piazzole di collegamento.

In ogni caso fate attenzione alla connessione dei diffusori onde evitare il controfase che altererebbe l'ascolto stereofonico.

Con il jack per le cuffie scelto per la motherboard, il circuito di rilevazione della cuffia provvederà a disabilitare l'uscita sugli altoparlanti; infatti tale jack ha l'interruttore che si apre inserendo lo spinotto. Nel caso non voleste questa funzione, dovreste



Fig. 1
Insieme di
motherboard
e modulo con
LM4835.



Fig. 2
Insieme della
motherboard
e del modulo
con PAM8403.

scegliere un jack differente (a tre soli elettrodi) e tenere il segnale marcato nello schema come HEAD_S connesso a massa per non interrompere l'uscita degli altoparlanti, ovvero farlo passare da un interruttore con il quale potrete scegliere manualmente se disattivare gli altoparlanti. □



Cosa occorre?

I componenti utilizzati in questo progetto sono di facile reperibilità, mentre l'amplificatore con PAM8403 (cod. MICROAMPLID2X3W) è venduto a Euro 4,00 da Futura Elettronica. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Non farti sfuggire nulla!

Riprendi ogni attimo con i mini registratori audio-video

SISTEMA DI VIDEOSORVEGLIANZA FULL HD OCCULTATO IN UN LIBRO

Libro con sistema di videosorveglianza occultato composto da telecamera Full HD e di microfono integrato. Dispone di sensore PIR per la rilevazione di movimento (funzione Motion Detection) che consente di attivare la registrazione solo in presenza di movimento. La telecamera è dotata di visione notturna e grazie alla tecnologia Starlight permette di effettuare riprese molto nitide anche in condizioni di scarsa luminosità. Le immagini riprese vengono salvate su una memoria SD card e possono essere trasferite su PC tramite il cavetto USB in dotazione. Integra una batteria da ben 8000 mAh che permette registrazioni continue fino a 30 ore. Il suo design curato minuziosamente in ogni particolare lo rende identico ad un libro vero e proprio, pertanto può essere collocato sia in casa che in ufficio, oppure tenuto in mano, senza destare alcun sospetto.



€ 64,00
cod. FR714

PENNE PER REGISTRAZIONE AUDIO-VIDEO, FOTO

Sembrano delle semplici penne ma nascondono un'efficace telecamera con la quale è possibile effettuare riprese audio-video e foto ad alta risoluzione. Ideali per uomini d'affari, investigatori, studenti, meeting, conferenze, colloqui; possono essere utilizzate anche come pen drive. Istruzioni in italiano.



€ 58,00
cod. FR707

CON MEMORIA INTERNA DA 8 GB

CON MEMORIA ESTERNA MAX 32 GB

€ 42,00
cod. FR708

BOTTIGLIETTA CON DVR E TELECAMERA

Occultato dietro l'etichetta di una vera bottiglia d'acqua da ½ litro, perfettamente utilizzabile con acqua da bere, è presente un compatto DVR con telecamera pinhole Full HD, microfono e batteria ricaricabile. Grazie al motion detection è possibile attivare la registrazione solo in presenza di qualcuno. Le immagini riprese vengono salvate su una micro SD card (acquistabile separatamente) e possono essere trasferite su PC tramite il cavetto USB in dotazione. Integra una batteria da ben 400 mAh che permette registrazioni continue fino a 3 ore.



€ 39,00
cod. FR737



Prezzi IVA inclusa

MINI DVR CON FISSAGGIO A CLIP

Mini DVR dotato di display LCD con fissaggio a clip. Estremamente leggero e compatto consente di registrare video e audio, scattare fotografie e memorizzare i file su SD card (acquistabile separatamente). La confezione comprende una clip per issare il mini DVR agli indumenti e un cavo USB per collegare il mini DVR al computer. Sul retro del dispositivo è presente un magnete che permette fissare il dispositivo ad un oggetto metallico. Batteria ricaricabile integrata. Istruzioni in italiano.



€ 46,00
cod. FR713



BRACCIALETTA SPORT CON TELECAMERA

Bracciale sport con telecamera integrata ad alta risoluzione. Oltre alla funzione di orologio consente di registrare video e audio, scattare fotografie e memorizzarle su SD card (acquistabile separatamente). Dispone inoltre di funzione webcam, funzioni sportive (monitoraggio del sonno, contapassi, consumo calorie) il tutto controllabile tramite smartphone (Android, iOS) via bluetooth. Batteria integrata di lunga durata da 350 mAh, ricaricabile tramite cavetto (incluso). Istruzioni in italiano.



€ 58,00
cod. FR706



OROLOGIO/SVEGLIA DA TAVOLO CON TELECAMERA NASCOSTA E DVR

Orologio/sveglia digitale da tavolo con telecamera CMOS Full HD incorporata dotata di 6 LED infrarossi per la ripresa notturna. È in grado di effettuare registrazioni audio e video (AVI) e salvarle direttamente su micro SD card max. 32 GB (acquistabile separatamente). Consente inoltre di scattare fotografie con una risoluzione di 2560 X 1440 pixel. Dispone di radiocomando per il controllo a distanza. Istruzioni in italiano.



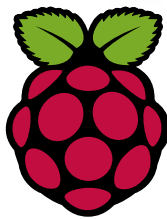
€ 69,00
cod. FR712





CCTV CON RASPERRY Pi E MOTIONEYE

di MARCO MAGAGNIN

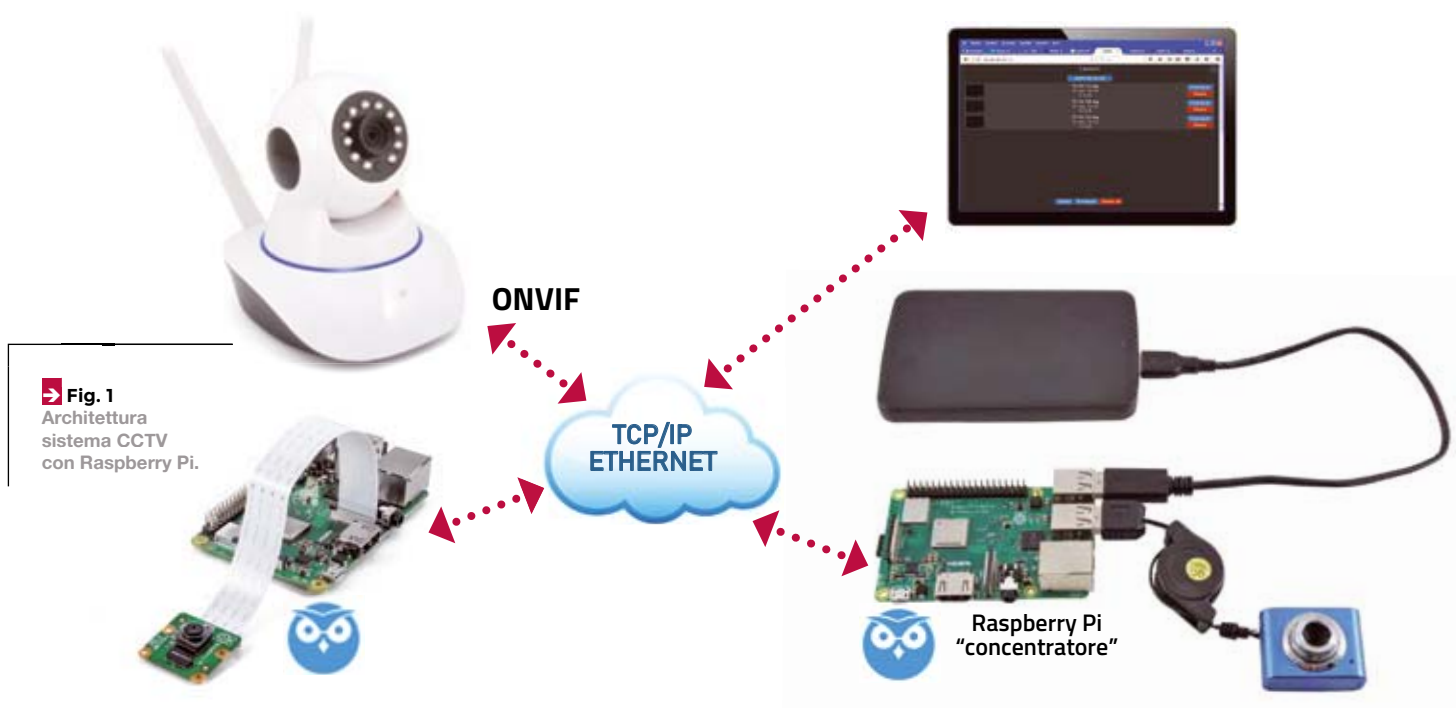


Costruiamo il nostro
sistema centralizzato
di videosorveglianza
domestico controllato
via web.
Ultima puntata

N

elle puntate precedenti di questa serie, dedicata alla realizzazione di un sistema di videosorveglianza basato su Raspberry Pi e sull'applicativo MotionEye, abbiamo visto come installare il software necessario alla nostra architettura e come configurare una prima videocamera

collegata via USB ed una seconda Camera Pi gestibile da remoto, utilizzando un secondo Raspberry Pi, raggiungibile dalla rete locale, con installato MotionEye. Abbiamo visto come configurare la gestione delle notifiche di allarme a fronte della individuazione di un movimento "sospetto" nell'area controllata dalla videocamera. A fronte di un evento abbiamo descritto la possibilità di inviare un messaggio via email e/o di eseguire un comando o un programma. Sempre con riferimento all'architettura che



riproponiamo nella **Fig. 1**, in questa puntata vediamo come ampliare la nostra architettura con una camera IP commerciale, il tutto sempre controllato dal nostro Raspberry Pi "concentratore". Premettiamo che le informazioni contenute in questo articolo sono di tipo "orientativo" per affrontare un progetto di questo tipo. Chi vuole riprodurre quanto descritto è opportuno che possieda una buona conoscenza del mondo GNU/Linux, del protocollo TCP/IP e della tecnologia SOAP (Simple Object Access Protocol – è la tecnologia adottata per lo sviluppo dei Web Services, categoria alla quale appartiene ONVIF). Ovviamente la conoscenza di un po' di programmazione nel linguaggio Python non guasta. La sfida, se così possiamo chiamarla, è stata di poter controllare dall'applicativo MotionEye una camera IP commerciale. È disponibile una vasta gamma di questa tipologia di videocamere, di buona qualità, con prezzi in continua discesa e dotate di connettività TCP/IP, possibilità di brandeggio e zoom PTZ (Pan, Tilt, Zoom). Generalmente, per l'utilizzo di queste videocamere,

vengono proposti software proprietari, ancorché gratuiti, di solito forniti come "app" per smartphone o applicazioni per PC. Fortunatamente, una tendenza sempre più diffusa è l'adozione da parte dei costruttori, del sistema operativo GNU/Linux e del protocollo di comunicazione ONVIF (Open Network Video Interface Forum) e del protocollo di streaming video RTSP (Real Time Streaming Protocol).

Troppe sigle? Procediamo con ordine.

La camera IP che abbiamo scelto è il modello IPC-Z06H commercializzata da Futura Elettronica (www.futurashop.it) con il codice CP728. Il problema principale che abbiamo incontrato è la mancanza di documentazione destinata ad un utilizzatore "maker", quindi abbiamo affrontato il progetto con un approccio di "reverse" engineering.

Osservando la camera IP, al di sotto abbiamo individuato un'etichetta con le credenziali di accesso al sistema di gestione della camera IP, come visibile in **Fig. 2**. Un primo indizio, meglio che niente.

Abbiamo collegato la camera IP alla rete locale via cavo Ethernet. Chiaramente la nostra rete locale prevede un server DHCP, pertanto alla nostra camera IP verrà assegnato in automatico un indirizzo IP appartenente alla nostra sottorete locale. Già ma quale indirizzo? Per scoprirlo avete due possibilità: o aprite l'interfaccia grafica del vostro router ed andate a cercare quali dispositivi

Fig. 2
Etichetta con
le credenziali
di accesso alla
Camera IP.



sono collegati al momento ed i loro rispettivi indirizzi IP, oppure utilizzate il comando *nmap* dalla shell di GNU/Linux, per esempio di Raspberry Pi. Nmap è uno strumento di mappatura di rete dalle numerosissime funzionalità, nel nostro caso lo utilizziamo per scoprire l'indirizzo IP della camera IP e su quali porte sono disponibili i servizi di rete. Se nmap non è installato, installatelo con i comandi:

```
sudo apt update
sudo apt install nmap
```

Poi digitate il comando:

```
nmap -p 1-10000 <indirizzo-IP-vostra-rete>/<CIDR>
```

Nmap è in grado di sondare le porte TCP aperte su ciascun device per individuare il tipo di servizio offerto.

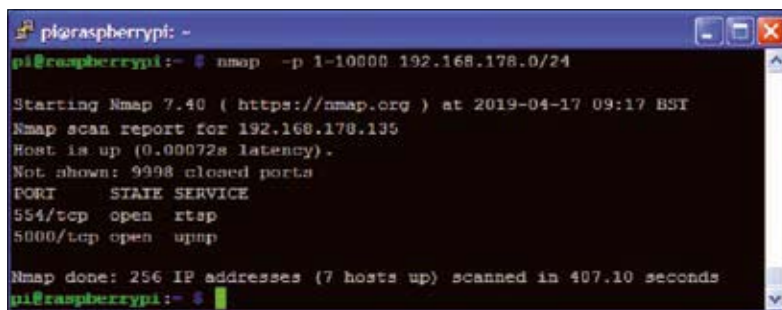
Il parametro *-p* permette di indicare l'intervallo di porte da scandire. Nel nostro caso da 1 a 10000. Normalmente non dovrebbero essere presenti servizi sulle porte superiori alla 10000.

Per <indirizzo-IP-vostra-rete> inserite l'indirizzo IP "network" della vostra sottorete locale, per esempio 192.168.178.0, per <CIDR> (Classless Inter-Domain Routing) inserite la lunghezza in bit della vostra maschera di sottorete, per esempio 24 per la maschera di uso comune "255.255.255.0".

Tra le numerose righe di uscita abbiamo individuato le seguenti, visibili in **Fig. 3**:

```
Nmap scan report for 192.168.178.135
Host is up (0.0017s latency).
Not shown: 9998 closed ports
PORT      STATE SERVICE VERSION
554/tcp    open  rtsp?
5000/tcp   open  upnp?
```

La nostra attenzione su questo dispositivo è stata attirata dalla presenza, sulla porta 554 di un servizio con protocollo RTSP, il protocollo di streaming video della nostra camera IP. A questo punto supponiamo che sulla porta 5000 sia presente il protocollo ONVIF. In aggiunta abbiamo anche l'indirizzo IP della nostra camera IP. Il nostro obiettivo è di poter intercettare lo streaming video per poterlo visualizzare tramite MotionEYE e, perché no, sempre da MotionEye poter gestire il brandeggio della camera IP. Non avendo alcuna



```
pi@raspberrypi: ~$ nmap -p 1-10000 192.168.178.0/24
Starting Nmap 7.40 ( https://nmap.org ) at 2019-04-17 09:17 BST
Nmap scan report for 192.168.178.135
Host is up (0.00072s latency).
Not shown: 9998 closed ports
PORT      STATE SERVICE
554/tcp    open  rtsp
5000/tcp   open  upnp
Nmap done: 256 IP addresses (7 hosts up) scanned in 407.10 seconds
pi@raspberrypi: ~$
```

Fig. 3
Risultato
scansione rete
locale.

documentazione al riguardo abbiamo deciso di chiedere "lumi" al protocollo "ONVIF" stesso. ONVIF è strutturato in base allo standard Web service, o "servizio web", che dir si voglia. Un servizio web è costituito da un insieme di messaggi che un dispositivo mette a disposizione per poter fornire e ricevere informazioni e comandi da parte di un utilizzatore terzo, in genere un altro dispositivo sulla rete che funge da utente. I messaggi ONVIF sono codificati sulla base delle specifiche SOAP (Simple Object Access Protocol) che è un protocollo per lo scambio di messaggi tra dispositivi, orientato agli oggetti, basato sul formato XML e che, in genere si appoggia a sua volta al protocollo HTTP per la trasmissione in rete. Per la nostra "indagine" abbiamo utilizzato lo strumento SoapUI (Soap User Interface), un applicativo che permette di "gestire" il protocollo SOAP con un'interfaccia grafica utilizzabile da un utente "umano". Le specifiche del protocollo ONVIF e delle diverse classi di messaggi sono documentate nel manuale "ONVIF Core Specification", scaricabile all'indirizzo:

<https://www.onvif.org/specs/core/ONVIF-Core-Specification-v250.pdf>

Non ci dilunghiamo sulla configurazione e preparazione di SoapUI per interrogare la camera PI. Questo richiederebbe ben più dello spazio di un articolo. Ci limitiamo ad illustrare i passi principali che abbiamo eseguito.

Qualche altra nota. Del protocollo ONVIF, al momento, esistono due versioni, la 1.0 (10) e la 2.0 (20). Le classi di messaggi ONVIF sono sviluppate con il linguaggio WSDL (Web Services Description Language) e sono disponibili sul sito di ONVIF. La prima classe di messaggi che andremo ad utilizzare è la classe "Device Management". Le specifiche sono reperibili all'indirizzo:

<https://www.onvif.org/ver10/device/wsd/devi->

cemgmt.wsdl

Una volta importata in SoapUI la troviamo come "Device Bindings".

Nella classe "Device Bindings" selezioniamo il messaggio (metodo) "GetServices". Nella finestra in alto potete vedere l'URL di destinazione del messaggio:

http://192.168.178.135:5000

Come potete vedere, è una richiesta HTTP all'indirizzo IP della camera IP sulla porta 5000, che abbiamo individuato con il comando "nmap". Modifichiamo il testo della richiesta inserendo "true" nel tag IncludeCapability.

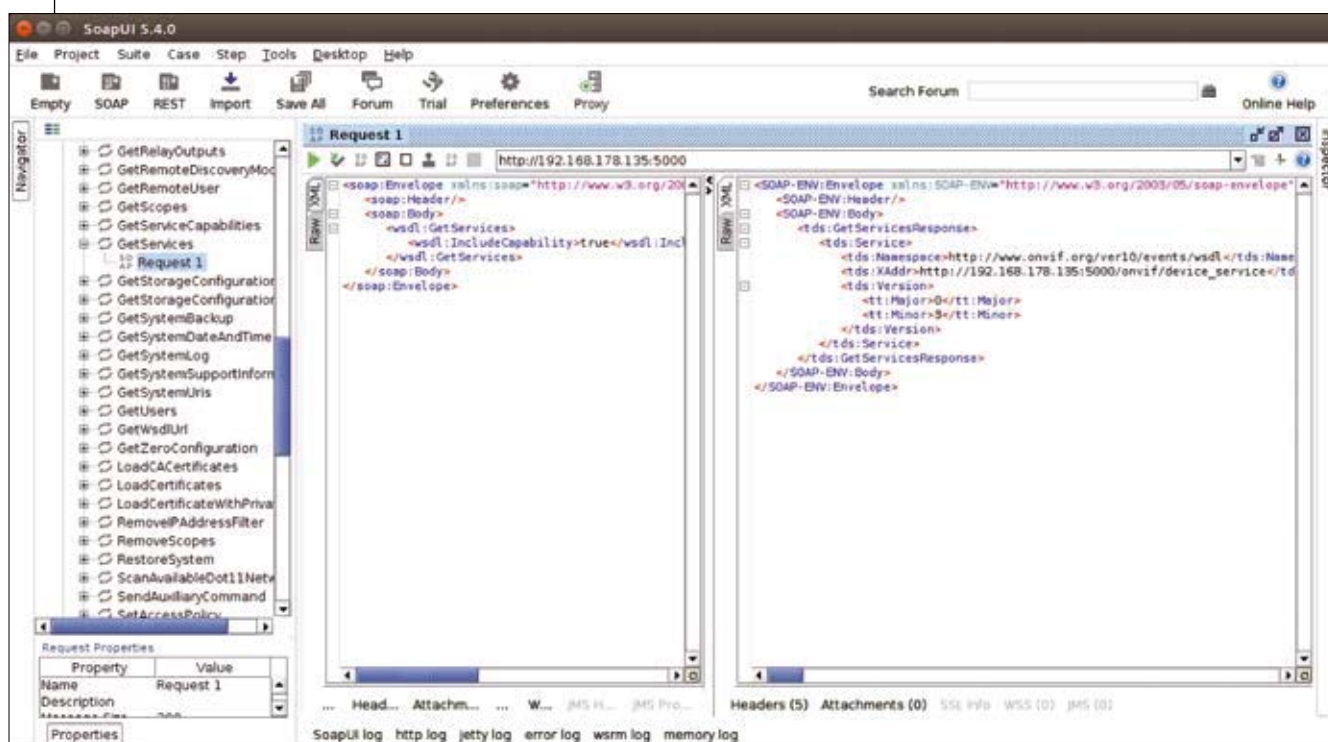
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wsdl="http://www.onvif.org/ver10/device/wsdl">
  <soap:Header/>
  <soap:Body>
    <wsdl:GetServices>
      <wsdl:IncludeCapability>true</wsdl:IncludeCapability>
    </wsdl:GetServices>
  </soap:Body>
</soap:Envelope>
```

Fig. 4
Risposta
messaggio
ONVIF
"GetServices".

Dopo aver premuto il pulsante "Submit" (triangolo verde in alto a sinistra) otteniamo la risposta:

```
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
  <tds:GetServicesResponse>
    <tds:Service>
      <tds:Namespace>http://www.onvif.org/ver10/events/wsdl</tds:Namespace>
      <tds:XAddr>http://192.168.178.135:5000/onvif/device_service</tds:XAddr>
      <tds:Version>
        <tt:Major>0</tt:Major>
        <tt:Minor>3</tt:Minor>
      </tds:Version>
    </tds:Service>
  </tds:GetServicesResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Purtroppo la risposta ci aiuta solo in parte. Infatti ci conferma che esiste un servizio ONVIF sulla porta 5000 corrispondente all'indirizzo della nostra camera IP ma non ci dice molto sulle altre classi supportate. Ci avrebbe fatto piacere trovare tra i Namespace il puntatore alla classe "media". O meglio alla sua versione. Dalle specifiche della camera IP sappiamo che supporta la versione 2.0



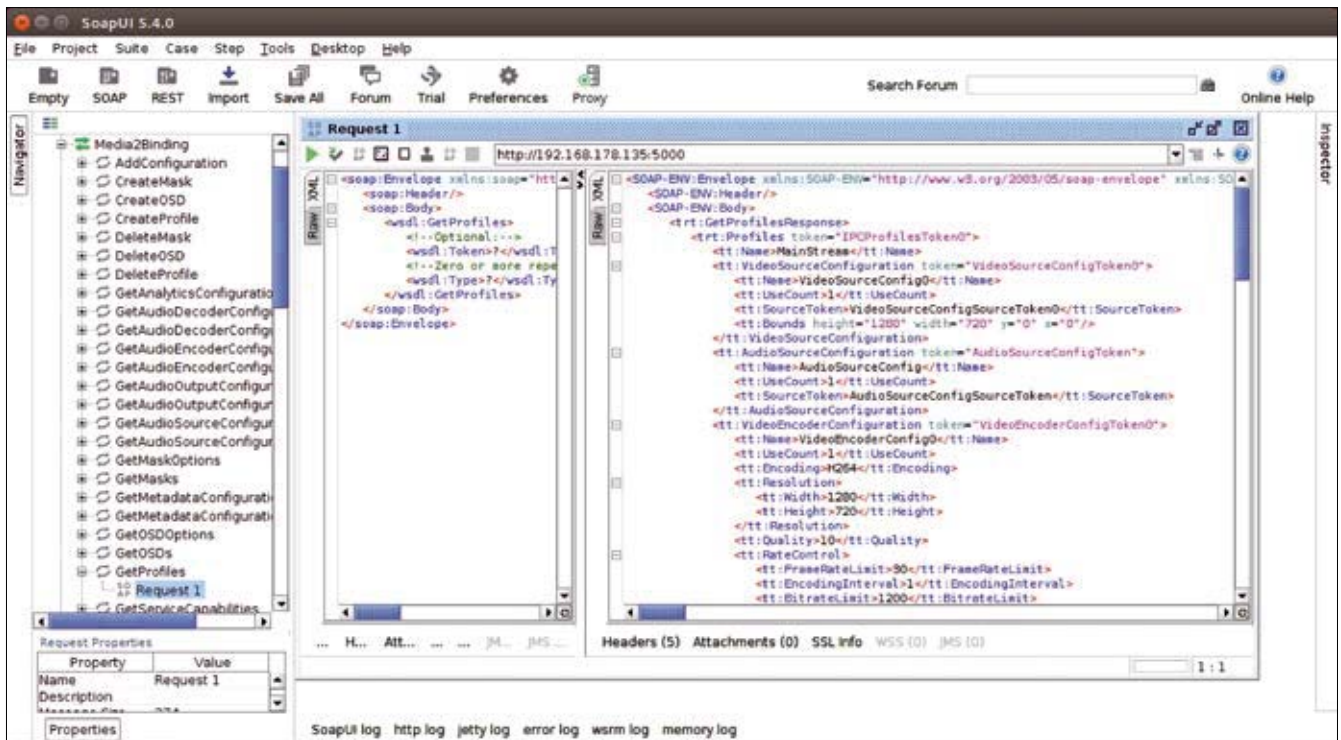


Fig. 5
Risposta
messaggio
ONVIF
"GetProfiles".

di ONVIF e quindi ci creiamo il nostro puntatore alla classe "media" in modo autonomo:

<https://www.onvif.org/ver20/media/wsdl/media.wsdl>

Ci interessano i messaggi della classe "media" perché dobbiamo individuare il nome del "Profilo" di configurazione definito dai costruttori della camera IP per personalizzare i messaggi ONVIF. A noi servirà per compilare correttamente i messaggi per la gestione del brandeggio (Pan/Tilt) della camera IP. Importiamo la classe "media" in SoapUI e configuriamo il messaggio "GetProfiles". Eseguiamo la richiesta ed otteniamo la risposta parzialmente visibile in **Fig. 5**.

```
<trt:GetProfilesResponse>
  <trt:Profiles token="IPCProfilesToken0">
    <tt:Name>MainStream</tt:Name>

    <tt:VideoEncoderConfiguration token="VideoEncoderConfigToken0">
      <tt:Name>VideoEncoderConfig0</
tt:Name>

      <tt:UseCount>1</tt:UseCount>
      <tt:Encoding>H264</tt:Encoding>
      <tt:Resolution>
        <tt:Width>1280</tt:Width>
```

```
<tt:Height>720</tt:Height>
</tt:Resolution>
<tt:Quality>10</tt:Quality>
<tt:RateControl>
  <tt:FrameRateLimit>30</
tt:FrameRateLimit>
  <tt:EncodingInterval>1</
tt:EncodingInterval>
  <tt:BitrateLimit>1200</
tt:BitrateLimit>
</tt:RateControl>
```

Abbiamo riportato nel testo le sezioni più importanti contenute nel messaggio di risposta. La prima è il nome del famoso "profilo di configurazione tanto cercato, nel nostro caso "IPCProfileToken0".

Nella seconda sezione possiamo vedere le informazioni di configurazione della telecamera, risoluzione, qualità, frame rate, ecc.

Nella stessa classe è disponibile anche il messaggio "GetStreamURI". La risposta a questo messaggio ci dice quali URI utilizzare per intercettare il flusso di streaming video della camera IP. Ci sarà utile per configurare la camera IP all'interno di MotionEye. Quindi, andiamo a curiosare. In Fig. 6 possiamo vedere la risposta del messaggio "GetStreamURI".

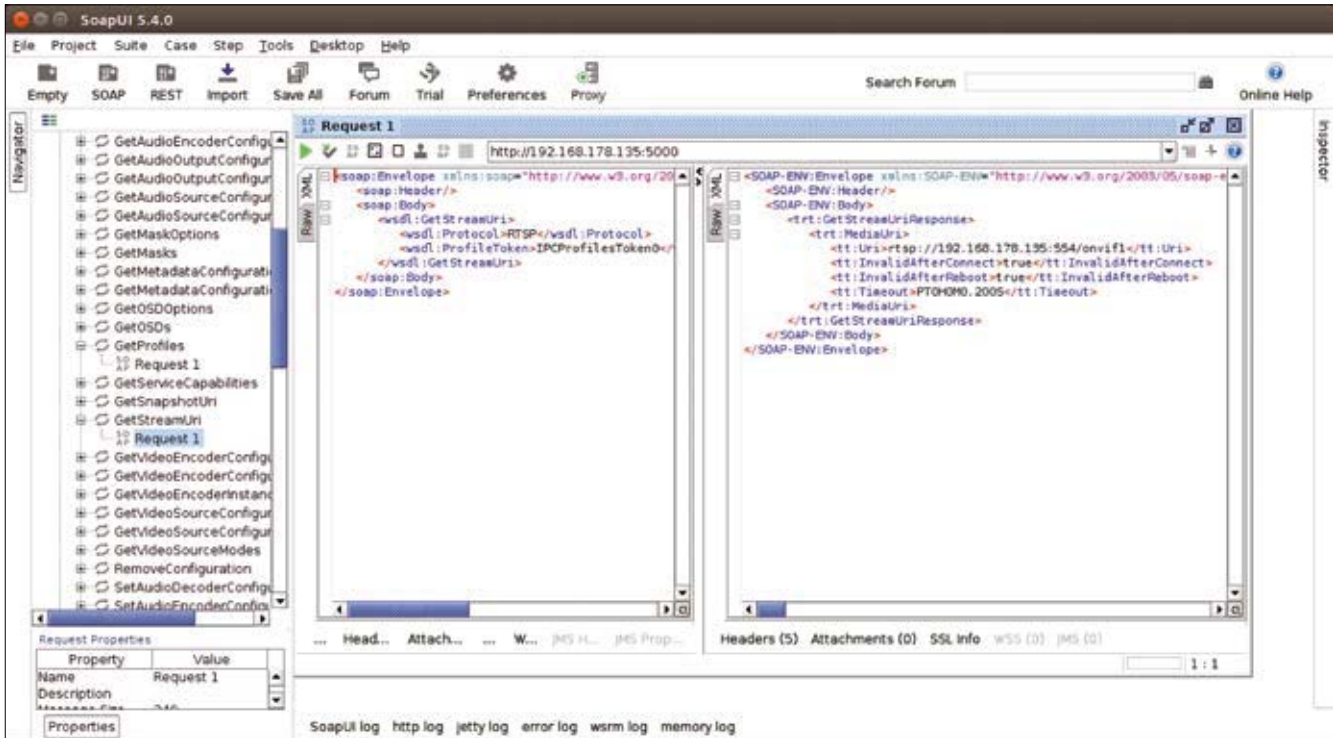


Fig. 6

Risposta
messaggio ONVIF
"GetStreamUri".

Riportiamo solamente l'indirizzo URI che ci interessa:

`rtsp://192.168.178.135:554/onvif1`

Per semplicità utilizzeremo il messaggio "ContinuousMove" che troviamo nella classe PTZ di ONVIF. La importiamo in SoapUI dall'indirizzo:

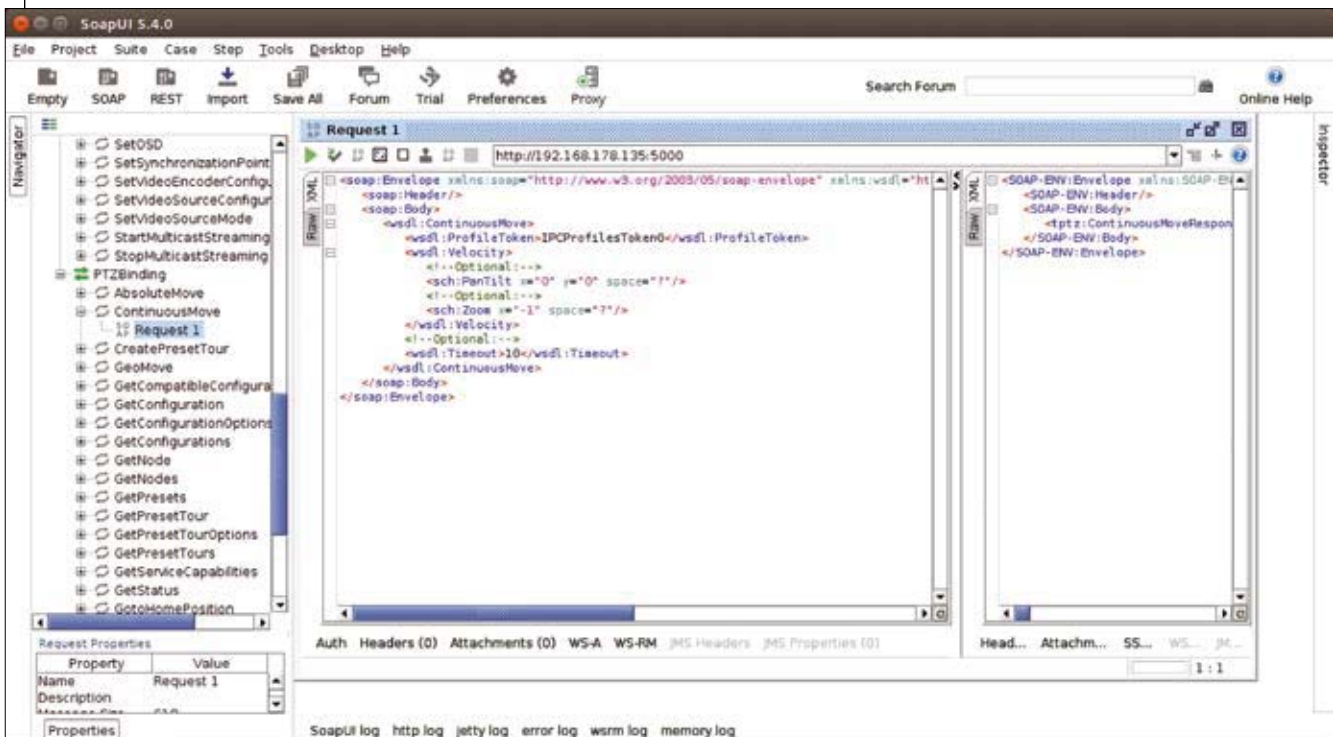
`https://www.onvif.org/ver20/ptz/wsdl/ptz.wsdl`

Fig. 7

Messaggio ONVIF.
"ContinuousMove".

Facciamo un'ultima verifica sul messaggio che permette di gestire il brandeggio della camera IP.

Vediamo il messaggio in **Fig. 7**.



Per indirizzarlo correttamente dobbiamo comunicare il Profilo, che abbiamo reperito in precedenza.

```
<wsdl:ContinuousMove>
  <wsdl:ProfileToken>IPCProfilesToken0</wsdl:ProfileToken>
  <wsdl:Velocity>
    <!--Optional:-->
    <sch:PanTilt x="0" y="0" space="?" />
    <!--Optional:-->
    <sch:Zoom x="-1" space="?" />
  </wsdl:Velocity>
  <!--Optional:-->
  <wsdl:Timeout>1</wsdl:Timeout>
</wsdl:ContinuousMove>
```

I parametri che ci interessano sono il "Profile-Token" che abbiamo reperito in precedenza, il parametro "PanTilt" ed il parametro Timeout. Il parametro PanTilt, nella sezione "Velocity" è un vettore che permette di impostare la velocità e la direzione di movimento del brandeggio. Il parametro "x" imposta il brandeggio orizzontale. I valori vanno da "-1", massima velocità verso sinistra a "+1", massima velocità verso destra. Il valore "0" significa nessun movimento. Similmente il parametro "y" imposta il brandeggio verticale. I valori anche in questo caso, vanno da "-1", massima velocità verso l'alto, a "+1", massima velocità verso il basso. Il valore "0" significa nessun movimento. Il parametro "Timeout" permette di configurare la durata del movimento. Con questi parametri potremo configurare un programma client scritto in Python che ci permetterà di "brandeggiare" la camera IP dall'interfaccia di MotionEye.

CONFIGURAZIONE MOTIONEYE PER CAMERA IP

Come abbiamo già descritto negli articoli precedenti il server web di MotionEye risponde all'indirizzo IP di Raspberry Pi sulla porta 8765. Pertanto, per accedere al vostro MotionEye dovete aprire un browser, per esempio Firefox, e digitare l'URL

<http://<indirizzo-IP>:8765>

dove l'indirizzo IP è quello del vostro Raspberry Pi "concentratore".

Vi apparirà la solita pagina di login visibile in **Fig. 8**. Ricordiamo che l'amministratore predefinito di MotionEye è "admin", senza password, quindi lasciate vuoto il campo password.



Fig. 8
Login di MotionEye.

Vi apparirà la pagina principale di MotionEye, (**Fig. 9**). Come noto i pulsanti in alto a sinistra permettono di configurare le videocamere collegate a MotionEye e di cambiare l'utente di lavoro. Il pulsante in alto a destra serve per uscire dall'applicazione.

Per aggiungere la camera IP, cliccate col tasto destro del mouse sul campo in alto a sinistra (vedi freccia rossa) e selezionate "Add Camera".

Vi viene proposta la finestra pop-up con le caratteristiche della videocamera che volete configurare.

Sono disponibili diverse possibilità. Per ciascuna si apre una modalità di configurazione specializzata, dove specificare percorsi e credenziali per accedere alla videocamera stessa. Nel nostro

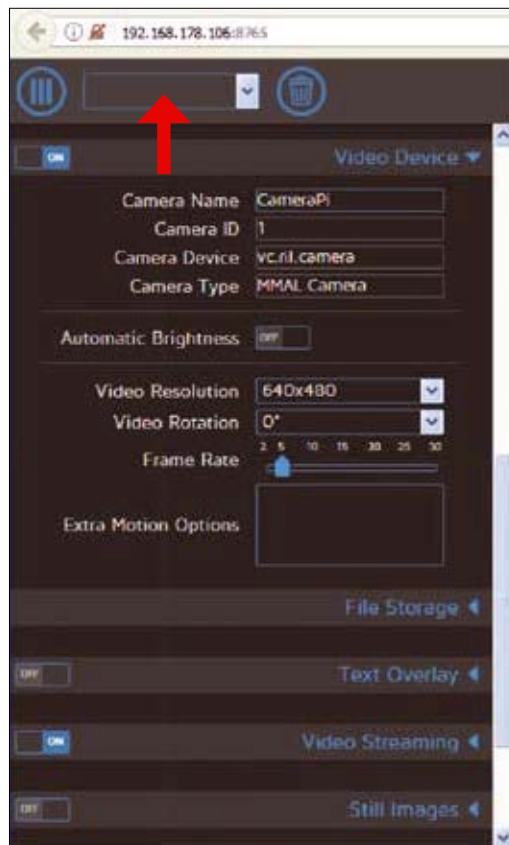


Fig. 9
Aggiunta camera IP.

Fig. 10
Descrizione camera IP remota.

caso, trattandosi di una Camera IP collegata via network selezionate "Network Camera" e come URL inserite quello reperito dal messaggio "GetStreamURI". Come Username e Password inserite quelle indicate sull'etichetta applicata alla camera IP o quella comunque indicata dal costruttore. Se tutto è andato per il verso giusto, la camera verrà riconosciuta come RTSP/TCP Camera (**Fig. 10**). Nel nostro caso, nel campo "Camera" abbiamo dovuto selezionare l'opzione "RTSP/UDP Camera".

Confermate e vi verrà presentato il menu con tutti le sezioni di configurazione disponibili. Nel frattempo potrete vedere le immagini riprese dalla camera IP, assieme alle altre videocamere

Fig. 11
Configurazione device video camera IP remota.

configurate, nella pagina principale di MotionEye. Nella sezione "Video Device" date un nome alla camera IP, nel nostro caso l'abbiamo chiamata "CameraIP". Prendete nota del numero che appare nel campo "Camera ID". Ci servirà in seguito per configurare la funzionalità di brandeggio. Lasciate disabilitate le altre sezioni di configurazione, a meno della sezione "Text Overlay". Non ha molto senso impostare l'intercettazione del movimento per una camera che è essa stessa a muoversi. Che ne dite?

INSTALLAZIONE PYTHON-ONVIF

E' tempo di occuparci del brandeggio della camera IP. MotionEye mette a disposizione una serie di "azioni" che possono essere utilizzate per eseguire delle operazioni aggiuntive. Ogni azione, se attivata, si materializza con la presenza di un pulsante (button) nella finestra relativa alla video-camera cui si vuole associare l'azione specifica.

- lock
- unlock
- light_on
- light_off
- alarm_on
- alarm_off
- up
- right
- down
- left
- zoom_in
- zoom_out
- preset1 to preset9

Per attivare un'azione è necessario inserire nella cartella di configurazione di MotionEye (/etc/motioneye) un file, il nome del quale è composto dal nome dell'azione seguito dal carattere di sottolineatura e dal "Camera ID" della videocamera alla quale si vuole associare l'azione stessa.

All'interno del file devono essere inseriti i comandi di shell (bash) relativi alle funzionalità che si vogliono associare all'azione. Per il brandeggio della camera IP utilizzeremo le azioni:

- up
- right
- down
- left

Per questo inseriremo nella cartella /etc/motioneye i file di scripting Bash: up_3, right_3,

down_3 e left_3. Allo startup MotionEye, vista la presenza di questi file nella sua cartella di configurazione, provvederà ad inserire i relativi pulsanti di attivazione, in basso a destra, nella finestra della camera IP. Per poter gestire il brandeggio della camera IP è necessario realizzare un client che permetta di colloquiare con i servizi web della camera IP stessa. Nel repository GitHub è disponibile il progetto "python-onvif" che è un'implementazione in Python di un client ONVIF. Il progetto è messo a disposizione, con licenza MIT, dalla Quatanium Technology Co. Ltd.,

che citiamo e ringraziamo per il lavoro svolto. Per installare python-onvif digitate i comandi:

```
sudo apt update
sudo apt dist-upgrade
```

Installate "suds", un client in python per gestire la messaggistica SOAP:

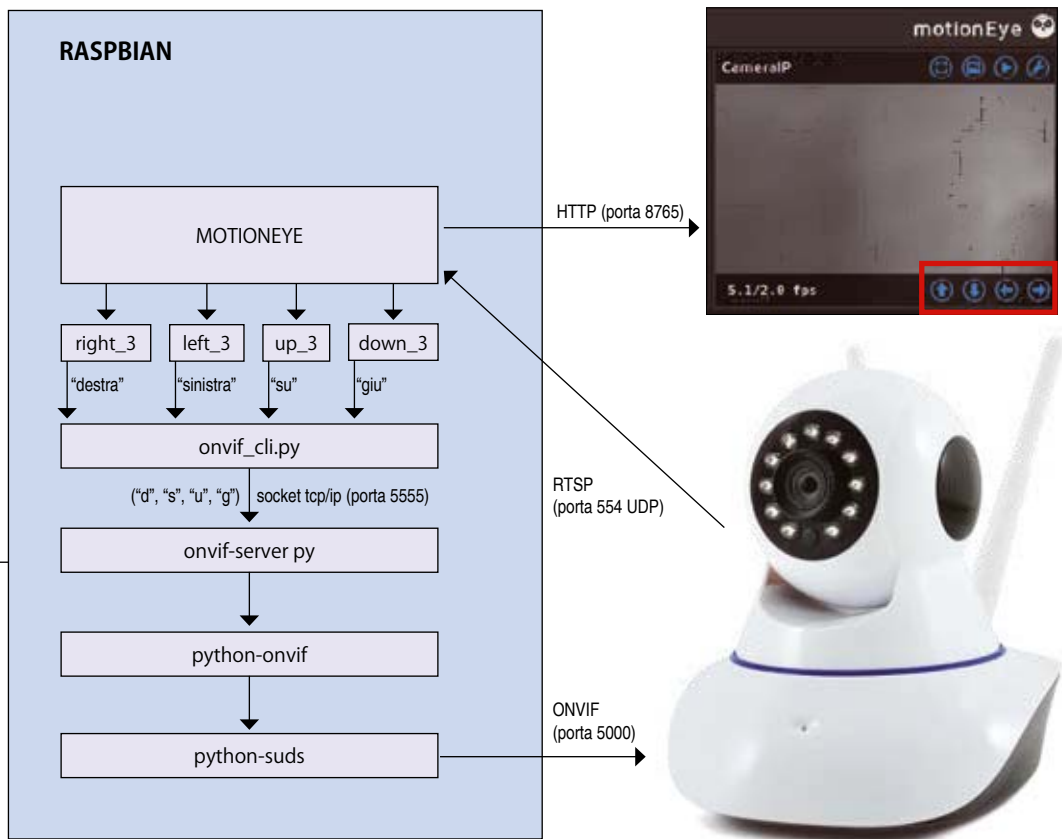
```
pip install 'suds>=0.4'
pip install suds-passworddigest
```

Installate python-onvif

↓ Listato 1 - onvif_server.py

```
#!/usr/bin/python
from onvif import ONVIFCamera
import sys
from socket import *
myHost = '192.168.178.119' # IP RPI concentratore
myPort = 5555
# Oggetto Camera con parametri connessione
mycam = ONVIFCamera('192.168.178.135', 5000, '15986051', '123')
# Direzione e velocità dei movimenti
XMAX = 1 # destra
XMIN = -1 # sinistra
YMAX = 1 # alto
YMIN = -1 # basso
tmov = 0 # durata movimento predefinita
# Crea oggetto ptz service
ptz = mycam.create_ptz_service()
request = ptz.create_type('ContinuousMove') # istanza messaggio ONVIF
request.ProfileToken = "IPCProfilesToken0" # Profilo camera IP
s = socket(AF_INET, SOCK_STREAM) # crea un socket TCP
s.bind((myHost, myPort)) # bind IP e porta
s.listen(5) # accetta 5 connessioni simultanee
while 1:
    # wait for next client to connect
    connection, address = s.accept() # accetta connessione da client
    while 1:
        comando = connection.recv(1) # riceve un carattere
        if (comando == 'd'): # destra
            print 'move right...'
            request.Velocity.PanTilt._x = XMAX
            request.Velocity.PanTilt._y = 0
            request.Timeout = tmov
        if (comando == 's'): # sinistra
            print 'move left...'
            request.Velocity.PanTilt._x = XMIN
            request.Velocity.PanTilt._y = 0
            request.Timeout = tmov
        if (comando == 'g'): # basso
            print 'move down...'
            request.Velocity.PanTilt._x = 0
            request.Velocity.PanTilt._y = YMAX
            request.Timeout = tmov
        if (comando == 'u'): # alto
            print 'move up...'
            request.Velocity.PanTilt._x = 0
            request.Velocity.PanTilt._y = YMIN
            request.Timeout = tmov
        # Invia messaggio ContinuousMove alla camera IP
        ptz.ContinuousMove(request)
        comando = ''
        break
    connection.close() # close socket
```


Fig. 12
Schema "server"
brandeggio camera IP.



`pip install onvif`
All'inizio abbiamo sperimentato la chiamata

diretta alla classe "ContinuousMove", ogni volta che veniva cliccato un pulsante i tempi di risposta erano decisamente troppo lunghi, dovuti in gran

↓ Listato 2 - onvif_cli.py

```
#!/usr/bin/python
import sys
from socket import *
serverHost = '192.168.178.119' # indirizzo IP server
serverPort = 5555             # porta server
comando = sys.argv[1]
print comando                 # solo per debug
muovi = 'x'
if (comando == 'destra'):
    print 'move right...'
    muovi = 'd'
if (comando == 'sinistra'):
    print 'move left...'
    muovi = 's'
if (comando == 'giu'):
    muovi = 'g'
if (comando == 'su'):
    print 'move up...'
    muovi = 'u'
s = socket(AF_INET, SOCK_STREAM) # create a TCP socket
s.connect((serverHost, serverPort)) # connect to server on the port
s.send(muovi)                     # invia richiesta al server
```

Listato 3 - right_3

```
#!/bin/bash
nohup python /home/pi/onvif_cli.py destra > /dev/null &
```

Listato 5 - up_3

```
#!/bin/bash
nohup python /home/pi/onvif_cli.py su > /dev/null &
```

Listato 4 - left_3

```
#!/bin/bash
nohup python /home/pi/onvif_cli.py sinistra > /dev/null &
```

Listato 6 - down_3

```
#!/bin/bash
nohup python /home/pi/onvif_cli.py giu > /dev/null &
```

parte alla creazione dell'oggetto ONVIFCamera (vedi **Listato 1**).
Di conseguenza abbiamo deciso di realizzare un programma "server" che inizializza una sola volta la classe ONVIFCamera e si mette in attesa su socket TCP/IP, porta 5555, di richieste da parte di un programma intermedio che viene richiamato

a sua volta dagli script attivati dai pulsanti. Nella **Fig. 12** è visibile l'architettura degli oggetti software preposti alla gestione del brandeggio della Camera IP. Quando viene premuto un pulsante di brandeggio nella videata della Camera IP viene eseguito lo script "Bash" corrispondente alla azione richiesta. Per esempio, in caso di movimento

Fig. 13
Aspetto
cartella /etc/
motioneye.

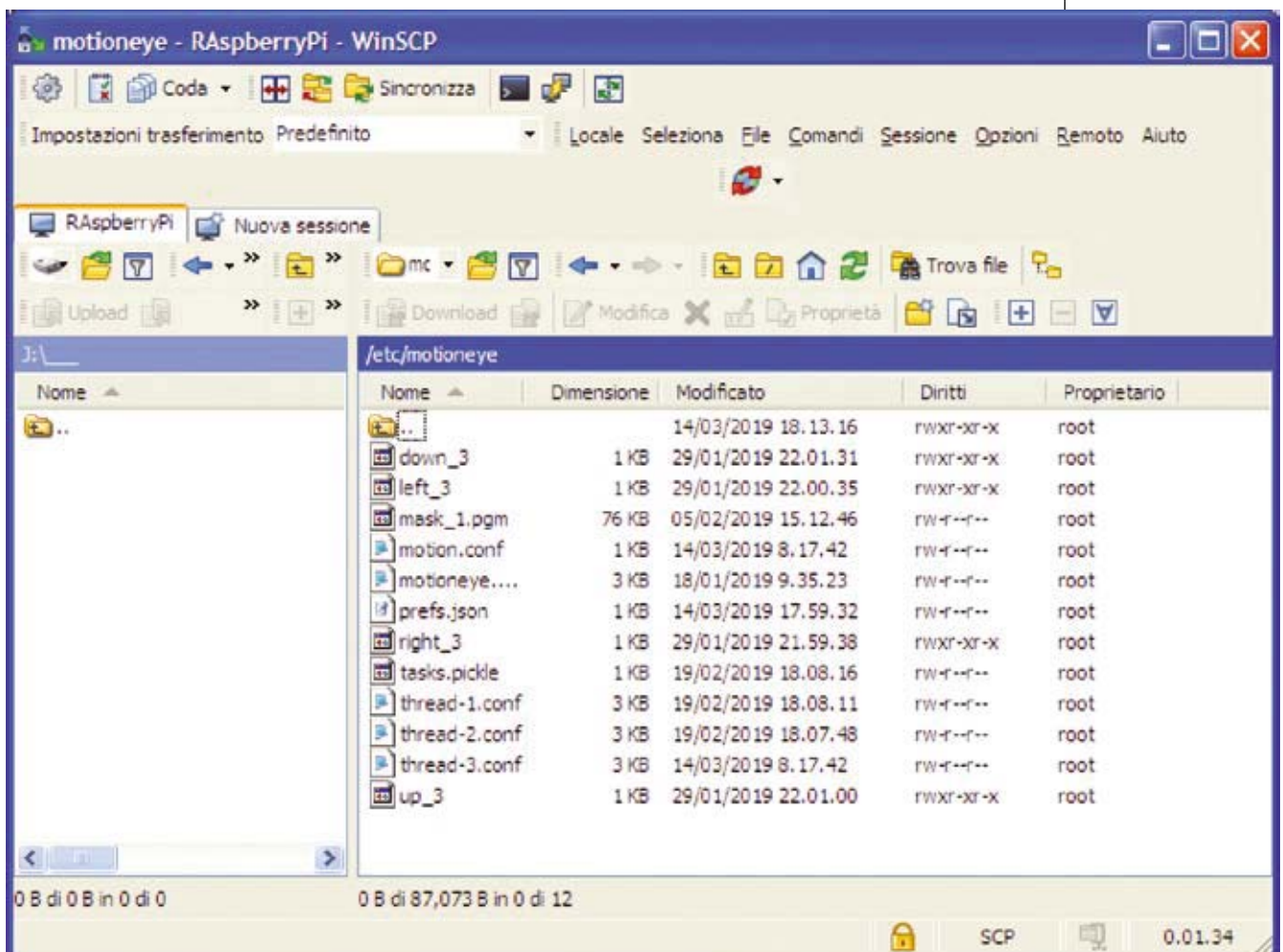




Fig. 14
Visione di insieme
delle immagini
riprese dalle
videocamere.

verso destra, verrà attivato lo script "right_3", che a sua volta richiama il programma "onvif_cli.py" passandogli il parametro "destra". Così per gli altri movimenti. Nei **Listati 3, 4, 5 e 6** potete vedere il semplicissimo contenuto di ciascuno script ovvero il richiamo del programma onvif_cli.py con il parametro che richiede l'esecuzione del rispettivo movimento della camera IP.

Il programma "onvif_cli.py" è visibile nel **Listato 2**. Quando richiamato da uno degli script associati ai pulsanti, analizza il contenuto del parametro passato da ciascuno script, confeziona una richiesta per il server "onvif_server.py", crea una sessione client via socket, sempre verso il server ed invia la relativa richiesta di brandeggio. A questo punto il server "onvif_server.py" compone il messaggio ONVIF corrispondente all'azione richiesta e lo consegna alla catena di componenti "python-onvif" e "python-suds" che provvedono alla trasmissione effettiva verso la Camera IP. Il risultato è il movimento della Camera nella o nelle direzioni richieste e il conseguente cambio di angolazione della immagine visualizzata.

Il messaggio di richiesta è di un solo carattere, il significato del quale è facilmente intuibile analizzando il **Listato 2**.

In **Fig. 13** potete vedere come deve presentarsi il contenuto della cartella /etc/motioneye.

Per lanciare il programma onvif_server.py automaticamente al boot di Raspbian inserite, nel file di configurazione dello schedulatore Cron, la direttiva per l'esecuzione del server.

Aprire il file di configurazione dello schedulatore Cron con il comando:

```
crontab -e
```

Alla fine del testo inserite la dichiarativa:

```
@reboot nohup python /home/pi/onvif_server.py > /dev/null &
```

Chiudete con <CTRL-x> e confermate. Eseguite un reboot e riaprite la pagina di MotionEye. Nel nostro caso è possibile vedere le immagini riprese dalle tre videocamere.

Per disporle come in **Fig. 14** utilizzate il cursore "Layout Rows" nella sezione di configurazione "Preferences". Cliccando su una immagine qualsiasi potete vedere tutti i pulsanti associati a ciascuna camera. Quelli da utilizzare per il brandeggio sono i quattro in basso a destra nell'immagine della CameraIP.

Approfondendo ancor di più le funzionalità disponibili ed utilizzando un disco fisso come memoria di massa per Raspberry Pi e, perché no, un UPS per garantire la continuità di alimentazione, possiamo realizzare un sistema di videosorveglianza integrato dalle caratteristiche praticamente professionali. □



Cosa occorre?

Il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. La board Raspberry Pi 3 Tipo B+ (cod. RPI3BPLUS) è in vendita a Euro 44,00, il modulo con telecamera da 8 megapixel per Raspberry Pi 3 (cod. 9132664RS) è disponibile a Euro 39,90. La telecamera è disponibile anche nella versione senza filtro IR (cod. 9132673RS) al prezzo di Euro 39,90. La telecamera IP WiFi (cod. CP728) è in vendita a Euro 54,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



€ 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.

Dai vita alle tue idee digitali con Micro:bit!

Micro:bit è una piccola scheda di sviluppo che integra un microcomputer a 32 bit; può essere utilizzata per imparare le basi della programmazione ed è adatta a tutte le età. Dispone di sensori di movimento, una bussola elettronica, un buon numero di I/O, alcuni pulsanti e numerosi LED disposti a matrice per lo sviluppo di interfacce grafiche. Grazie al bluetooth integrato è possibile programmarla facilmente dal web, da PC, tablet e smartphone.



Impara ad utilizzarla con il kit Smart Robot Car...



Il kit permette di realizzare un piccolo robot con ruote controllato tramite telecomando IR. Una volta programmato può inseguire una linea, evitare gli ostacoli, seguire un oggetto, accendere i LED RGB e tanto altro. La confezione comprende la scheda micro:bit, una breakout per micro:bit, due micro motoriduttori con ingranaggi in metallo, due ruote e due staffe di supporto, un ball caster, un sensore ad ultrasuoni con base di supporto e staffe di fissaggio, un cavo USB-micro USB, un telecomando a infrarossi, un piccolo cacciavite, quattro distanziali femmina-femmina M3x40 e due maschio-femmina M2x10, una batteria al litio da 3,7 volt, due adesivi in velcro, viti e dadi per il montaggio.

Cod:

MICROBITROBOT

€ 109,00

e con il prototyping kit!

Il kit comprende: la scheda Micro:bit, una scheda con connettore per Micro:bit, un cavo USB-microUSB, un portabatterie per 2 batterie da 1,5 volt tipo AA, due batterie da 1,5 volt tipo AA, una breadboard a 400 contatti, 1 LED RGB, un Buzzer da 5 volt, jumper, mini pulsante, 3 LED (1 rosso, 1 verde, 1 giallo), resistenze di diversi valori e una fotoresistenza.



Cod: **MICROBITKIT**

€ 39,90

Incluso nel kit
IL NUOVO MANUALE
a colori che spiega nel dettaglio
come utilizzare e programmare micro:bit

CELLE DI CARICO

di DAVIDE SCULLINO

Conosciamo i sensori e l'elettronica necessaria per la misura del peso di oggetti in moltissimi ambiti, dalle bilance elettroniche alle pesche industriali.

Q

Quando si parla di misura del peso di un oggetto, s'intende il rilevamento della forza che questo esercita sul sistema di rilevamento, qualsiasi esso sia, per effetto della gravità terrestre (forza di gravità); ecco perché in Fisica si distingue tra peso e massa, giacché quest'ultima è la consistenza di un corpo, mentre il peso è la percezione che la massa trasmette sotto forma di forza esercitata per effetto dell'accelerazione di gravità. Non a caso massa e peso sono correlate dall'accelerazione di gravità, che mediamente vale $9,81 \text{ m/s}^2$. Da questo discorso deriva che misurare un peso significa rilevare una forza, che non a caso chiamiamo forza-peso. Siccome siamo elettronici, a noi interessano principalmente i dispositivi di misura del peso di tipo elettronico e tra questi

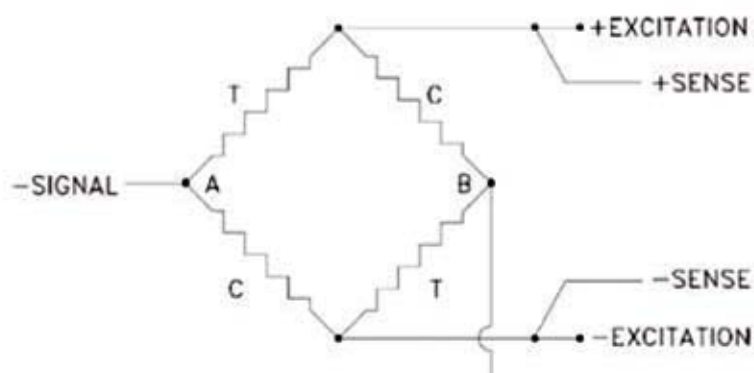


Fig. 1
Tipica
configurazione
elettrica degli
estensimetri nelle
celle di carico.

abbiamo i sensori di pressione (che sono effettivamente in grado di rilevare una forza, seppur per unità di superficie) e le celle di carico, le quali sono attualmente i sensori d'elezione per rilevare il peso in moltissimi ambiti, dalle bilance elettroniche alle pesche industriali.

Una cella di carico è un dispositivo che viene utilizzato per misurare il peso e, di riflesso, le forze esercitate, tanto a compressione, quanto a trazione. In questo articolo spiegheremo di cosa si tratta e in che modo si usa tale dispositivo.

COME FUNZIONA LA CELLA DI CARICO

Quando una forza viene applicata ad essa in un modo specifico, una cella di carico produce un segnale di uscita che è proporzionale alla forza stessa. Ma come fa a generare questo segnale? Ebbene, il dispositivo si basa su componenti detti *estensimetri*, che funzionano sfruttando l'effetto piezoresistivo, ossia la tendenza che alcuni materiali elettricamente conduttori hanno a modificare la propria resistività quando vengono compressi o espansi lungo certe direzioni. Tale fenomeno è qualcosa di simile all'effetto piezoelettrico che si manifesta nei quarzi e in alcuni materiali ceramici sintetici, dove la trazione o compressione produce

una differenza di potenziale tra due facce opposte. Una cella di carico del tipo strain-gauge (a estensimetro) è costituita da un elemento elastico che viene selettivamente indebolito per creare regioni relativamente ad alta tensione; in tali regioni si applicano gli estensimetri. Questi vengono collegati in varie configurazioni, sebbene la più apprezzata sia quella a Ponte di Wheatstone, formato da quattro estensimetri disposti come mostrato nella **Fig. 1**; in alcuni casi gli estensimetri sono due e gli altri due elementi sono resistenze comuni. Gli estensimetri possono anche essere disposti a formare mezzo ponte e in questo caso si parla di semicelle.

Il ponte di Wheatstone viene alimentato tra i nodi superiore e inferiore (punti +/- EXCITATION) da una tensione continua e stabilizzata di eccitazione, mentre il segnale d'uscita si preleva fra i nodi orizzontali del ponte; siccome il ponte di Wheatstone è sostanzialmente un doppio partitore resistivo di cui si prelevano i potenziali d'uscita sollevati da massa, la tensione che ne deriva è la differenza tra essi. Nello schema di **Fig. 1**, l'uscita è localizzata tra A e B.

Senza alcun carico applicato alla cella, tutti gli estensimetri hanno la stessa resistenza e quindi non c'è differenza di potenziale tra i punti A e B. Quando il carico viene applicato alla cella, la resistenza degli indicatori di tensione aumenta, mentre quella degli indicatori di compressione diminuisce, cosicché il ponte diventa sbilanciato e si verifica una differenza di potenziale proporzionale al carico applicato, che può essere misurata attraverso i punti A e B.

Alcune celle di carico e indicatori dispongono di cavi di rilevamento che nella predetta **Fig. 1** sono chiamati + e - SENSE e che consentono all'indicatore di misurare e regolare la tensione di eccitazione applicata alla cella; ciò è particolarmente importante quando per collocazione, le celle devono essere collegate con dei cavi lunghi. Le celle di carico possono essere utilizzate singolarmente o in combinazioni a seconda dalla geometria dell'oggetto da pesare.

La **Fig. 2** illustra un tipico estensimetro metallico, che consiste in una lamina metallica incisa su linee di griglia parallele, che formano un circuito delimitato dalle piazzole di uscita; la lamina è legata ad un materiale di supporto isolante che, a sua volta, è vincolato alla superficie della cella di carico, come appare nella stessa **Fig. 2**.

Nelle celle di carico a estensimetri, un'estremità di una cella di carico è tipicamente fissata a una

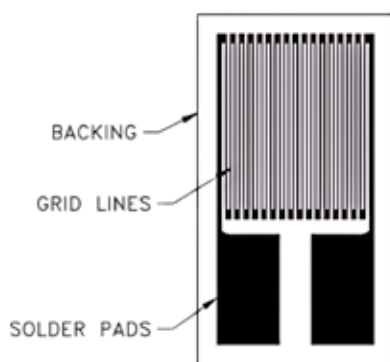


Fig. 2
Schematizzazione
dell'estensimetro
per celle di carico.

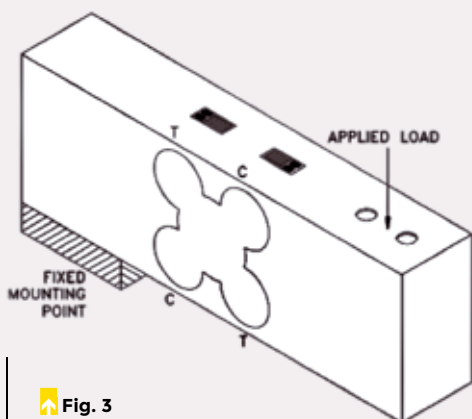


Fig. 3
Cella di carico a
trave con quattro
estensimetri.

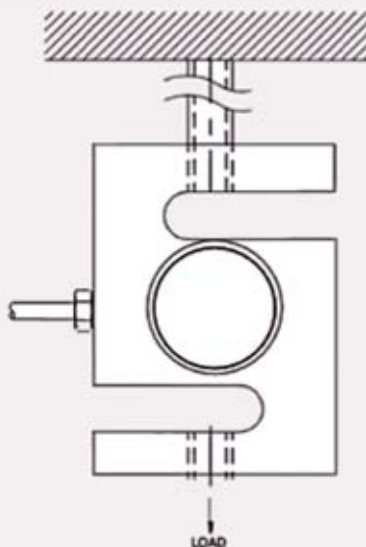


Fig. 4
Cella di carico a
S funzionante a
trazione.

struttura rigida, mentre l'altra è applicata a un dispositivo di ricezione del carico attraverso il quale viene applicato il carico o la forza da misurare; in questo caso si tratta di una cella a doppio punto, che possiamo assimilare a una trave a sbalzo, incastrata ad un estremo e sollecitata all'altro. Questo tipo di cella di carico è dei più comuni per la sua semplicità e il basso costo. La struttura è spesso binoculare, ossia una trave con due fori. Gli estensimetri sono montati sulla parte superiore ed inferiore per misurare le forze di trazione e di compressione. Poiché gli estensimetri possono essere danneggiati, sono in genere coperti e riparati da un soffietto in gomma. La trave stessa spesso è in lega di acciaio nichelata; per carichi leggeri (fino a 230 grammi), la trave è in rame-berillio, mentre per carichi maggiori è in acciaio inossidabile. In questi modelli, gli estensimetri tipicamente sono protetti da un rivestimento in uretano.

Nella **Fig. 3** vediamo una cella di carico a trave che riporta i quattro estensimetri di cui dispone, siglati T e C e disposti, incrociati, sulle due facce della

struttura metallica indebolita nella zona centrale per favorirne la deformazione. Con la disposizione illustrata, due estensimetri misurano lo sforzo di trazione e due rilevano lo sforzo a compressione, esattamente come avviene quando un carico (peso) è applicato alla cella di carico. Essendo quattro, gli estensimetri sono collegati insieme a formare un ponte di Wheatstone come illustrato nella già vista **Fig. 1**, però in questo caso la loro disposizione è tale da renderli collaboranti, altrimenti la variazione di resistenza negli uni andrebbe a contrastare quella negli altri; collaboranti significa che i due T sono opposti nel ponte, in modo da ottenere lo sbilanciamento all'applicare della forza.

Una tensione di eccitazione stabilizzata viene applicata agli angoli opposti del ponte di Wheatstone e un segnale viene misurato sugli altri, i punti A e B.

Oltre al tipo proposto in **Fig. 3**, che lavora a taglio, esiste quello mostrato nella **Fig. 4**, che lavora a trazione (cella a tensione) e viene utilizzato ad esempio nella pesatura di serbatoi e tramogge,

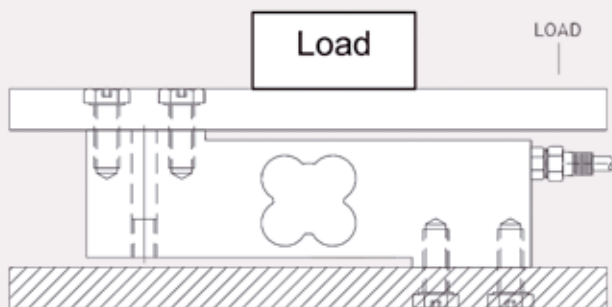
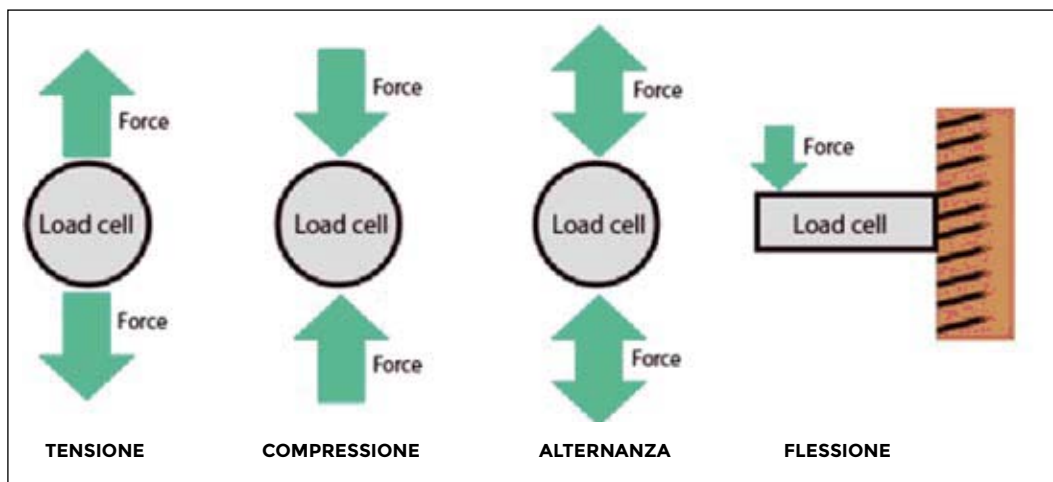


Fig. 5
Cella di carico a
punto singolo: il
peso da rilevare
viene applicato
direttamente
sopra.

➔ **Fig. 6**
Classificazione
delle celle di
carico per tipo di
sollecitazione.

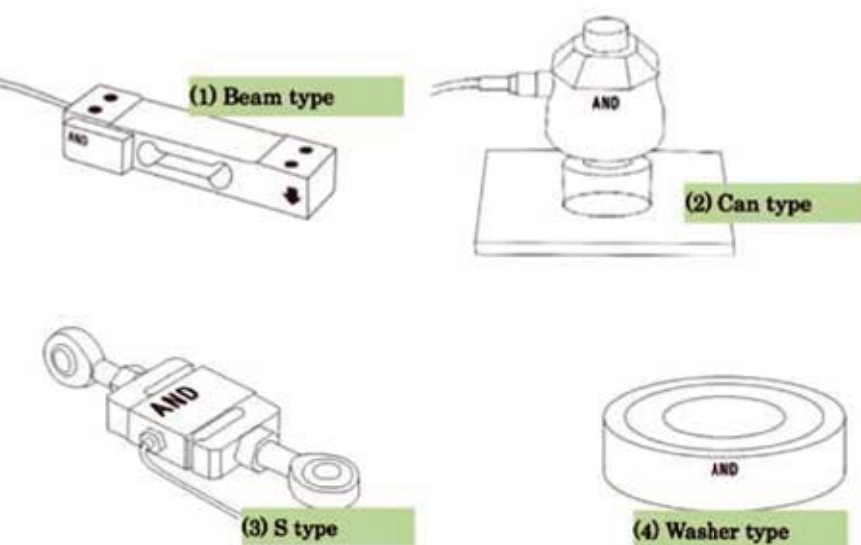


dove le celle di carico sono sospese da una struttura aerea e l'oggetto da pesare viene appeso dal lato inferiore. Tale struttura è praticamente una S e viene ampiamente utilizzata nella conversione di bilance meccaniche in elettromeccaniche. In questa situazione la cella viene utilizzata per rilevare la tensione dell'asta che collega il sistema a leva al raggio.

Le capacità tipiche per questo genere di cella vanno da 45 kg a 22 tonnellate, mentre quelle a trave spaziano da circa 500 g a decine di kg.

Esistono poi le celle di carico a punto singolo (**Fig. 5**) che sono ampiamente utilizzate nelle bilance a piattaforma; rilevano il peso indipendentemente da dove il carico è applicato alla piattaforma superiore. Le capacità variano tipicamente da 1 kg a 2 tonnellate e possono ospitare piattaforme aventi superficie di diverse centinaia di cm².

↓ **Fig. 7**
Cella di carico a
S funzionante a
trazione.



TIPOLOGIE DI CELLE DI CARICO

È possibile fare una classificazione delle celle di carico in base alla direzione di applicazione della forza, quindi abbiamo i tipi a *tensione*, *compressione*, *alternanza* e *flessione* (**Fig. 6**): il primo è tipicamente a S, mentre il secondo è a bottone o barilotto; il terzo lavora sia a trazione che a compressione e il quarto è a trave.

Esiste anche una classificazione per precisione della misura: Ultra-preciso, Preciso, Standard. La forma del materiale della molla dipende dalle caratteristiche della cella di carico.

Un'altra classificazione può essere fatta per tenuta all'aria, secondo quanto esposto qui di seguito.

- **Sigillato**: questo tipo racchiude le aree degli estensimetri all'interno di una custodia e li protegge dall'ambiente. In generale, l'interno del case è pieno di gas inerte, il che rende tali dispositivi idonei all'utilizzo in condizioni ambientali sfavorevoli, come all'esterno, in presenza di polveri o forte umidità. Un diaframma o un soffiato viene utilizzato in modo da non influire sulla piegatura del materiale della molla.
- **Aperto**: in esso gli estensimetri non sono protetti o al più vengono riparati da della resina morbida o gomma, che protegge dagli agenti ambientali. È idoneo a operare in ambienti dove la temperatura non sia troppo elevata e non vi sia eccessiva umidità o polvere.
- **Explosion-Proof**: grazie alla sua struttura questa tipologia di cella è a prova di fuoco.

Oltre che per composizione, sollecitazione e tenuta, le celle di carico possono essere classificate in base alla loro forma (**Fig. 7**); abbiamo quindi Beam Type (a trave), Can Type (a barilotto), S Type (ad S) e Washer Type (a rondella o bottone).

Passiamo ora al secondo integrato di quelli designati alla lettura dei segnali delle celle di carico, il cui schema a blocchi interno con applicazione tipica potete vedere nella Fig. 9. Nella Fig. 10 vedete invece la relativa pin-out.

L'integrato INA125 è un amplificatore per piccoli segnali ad alta precisione e a basso consumo, che integra un riferimento di tensione di elevata precisione. È l'ideale per l'abbinamento alle celle di carico e alla loro gestione, intesa come condizionamento del segnale prodotto nella configurazione a ponte di Wheatstone, in quanto dispone di un amplificatore differenziale all'ingresso e di un'uscita a tensione ben stabilizzata per alimentare le estremità del ponte stesso. Un resistore singolo ed esterno imposta il guadagno dell'integrato, che può essere compreso tra 4 e 10.000 volte in tensione. L'INA125 ha una ridottissima tensione di offset, una deriva termica dell'offset contenuta ($2\mu\text{V}/^\circ\text{C}$) e un'elevata

reiezione del rapporto in modo comune particolarmente elevata (100dB con $G=100$). Può funzionare alimentato a tensione singola di valore compreso fra 2,7V e 36V o ad alimentazione duale (da $\pm 1,35\text{V}$ a $\pm 18\text{V}$). Il riferimento di tensione è estremamente regolabile con le tensioni selezionabili dai pin di 2.5V, 5V o 10V, permettendo l'uso di una varietà di trasduttori. La tensione di riferimento è a 0.5% (massima) con una deriva di $\pm 35\text{ppm}/^\circ\text{C}$ (massima). La modalità "sleep" permette l'arresto e l'operazione di duty-cycle per risparmiare potenza.

L'INA125 è disponibile in DIP in plastica a 16 pin e confezioni SO-16 a montaggio superficiale ed è specificato per il range di temperatura compreso tra -40°C e $+85^\circ\text{C}$.

Le caratteristiche dell'INA 125 sono le seguenti:

- bassa corrente di quiescenza: $460\mu\text{A}$;
- riferimento di bassa tensione: 1,24V - 2,5V - 5V o 10V;

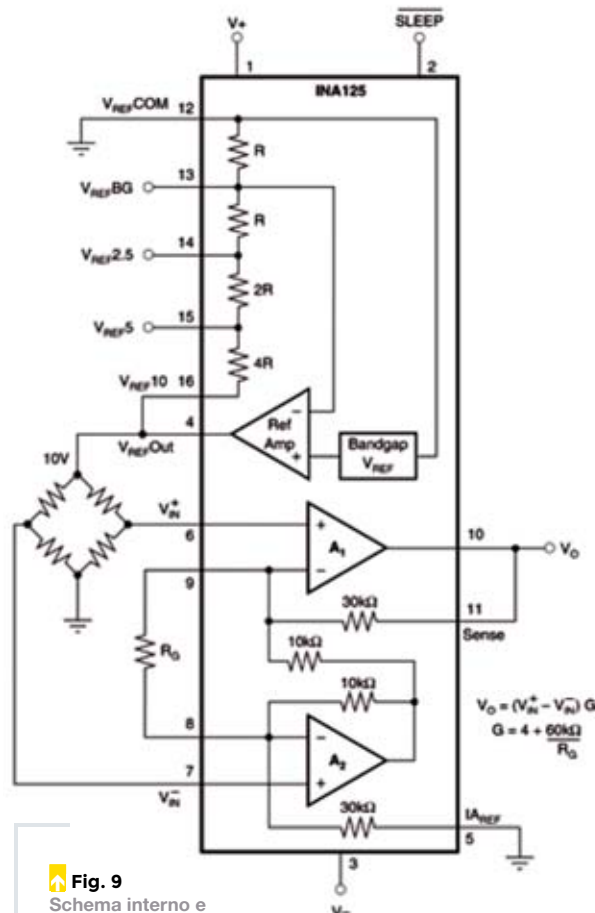


Fig. 9
Schema interno e
tipica applicazione
dell'INA125.

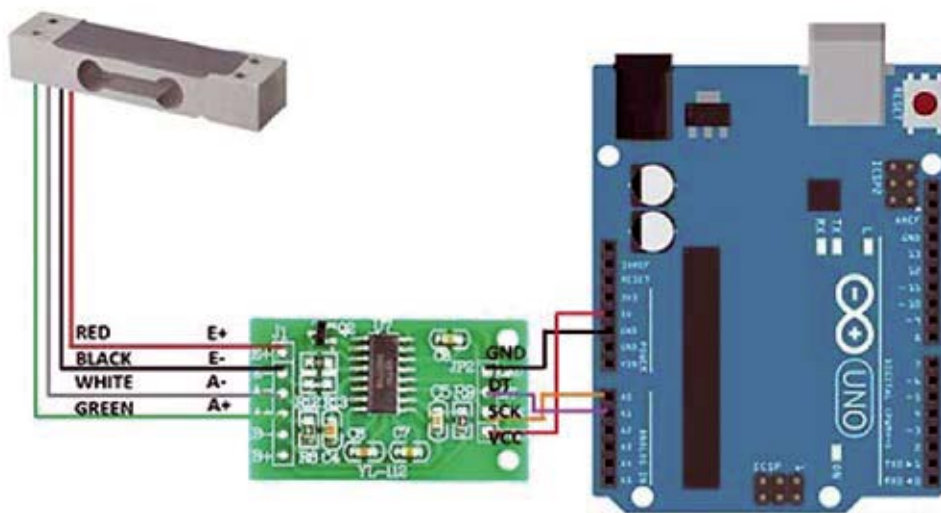
di carico a estensimetri: ad esempio INA125 o HX711; le caratteristiche di entrambi sono riportate nei riquadri a essi dedicati in queste pagine.

CELLE DI CARICO E ARDUINO

Naturalmente il pubblico di Arduino trova inte-

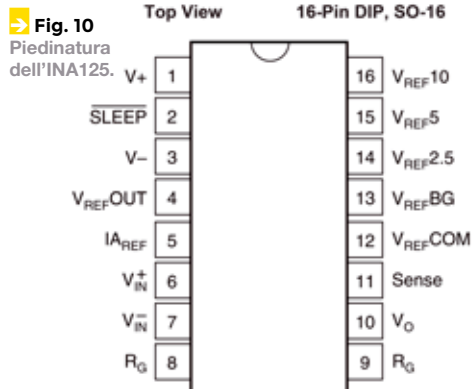
ressante le celle di carico ed è per questo che abbiamo pensato di realizzare qualche semplice applicazione da proporre; va subito precisato che per leggere con Arduino il segnale offerto dalle celle di carico occorre interporre un circuito di condizionamento ed allo scopo ben si prestano i

Fig. 11
Connessione di
una cella di carico
ad Arduino tramite
la breakout board
HX711.



- modalità "sleep";
- tensione di offset bassa: 250 μV max;
- deriva a basso offset: $2\mu\text{V}/^\circ\text{C}$ max;
- bassa corrente di polarizzazione di offset: 20nA max;
- rapporto di reiezione di modo comune (CMR): 100 dB min;
- rumore minimo: $38\text{nV}/\sqrt{\text{Hz}}$ con $f=1\text{kHz}$;
- protezione d'ingresso a $\pm 40\text{V}$;
- alimentazione singola: 2,7V a 36V;
- alimentazione duale: da $\pm 1,35\text{V}$ a $\pm 18\text{V}$;
- 16 pin DIP e pacchetti SO-16 SOIC.

Tra le applicazioni dell'integrato troviamo amplificatori di ponte di pressione e temperatura, controlli di processi industriali, automazione, acquisizione di dati multi-canale, strumentazione generica ecc.



due integrati appena descritti. Nelle applicazioni descritte utilizzeremo una breakout che monta l'HX711 (cod. HX711BOARD) e le celle di carico da 500g della Futura Elettronica (cod. LO-ADCELL500G) che collegheremo ad Arduino come mostrato nella **Fig. 11**.

Come vedete, la cella dispone di quattro fili e si connette come mostrato. Dall'uscita della breakout board, la quale trae alimentazione dal 5V di Arduino (e a sua volta alimenta la cella di carico) viene prelevato il segnale digitale contenente l'informazione sulla forza peso. Più esattamente, la breakout board fornisce stringhe di dati su un bus a due fili tipo I²C attraverso la linea DT (dati) sincronizzate con il clock che Arduino genera sulla linea SCK. Il **Listato 1** propone il codice necessario ad acquisire le letture effettuate tramite la cella di carico e l'HX711.

Passiamo adesso a un secondo esempio applicativo basato su una configurazione differente, ossia 2 semicelle carico con portata di 50kg l'una (cod. LC50KG), per una portata totale pari a 100kg. Elettricamente, queste mezze celle costituiscono

Fig. 12
Utilizzo dell'HX711 con due semi-celle da 50kg.

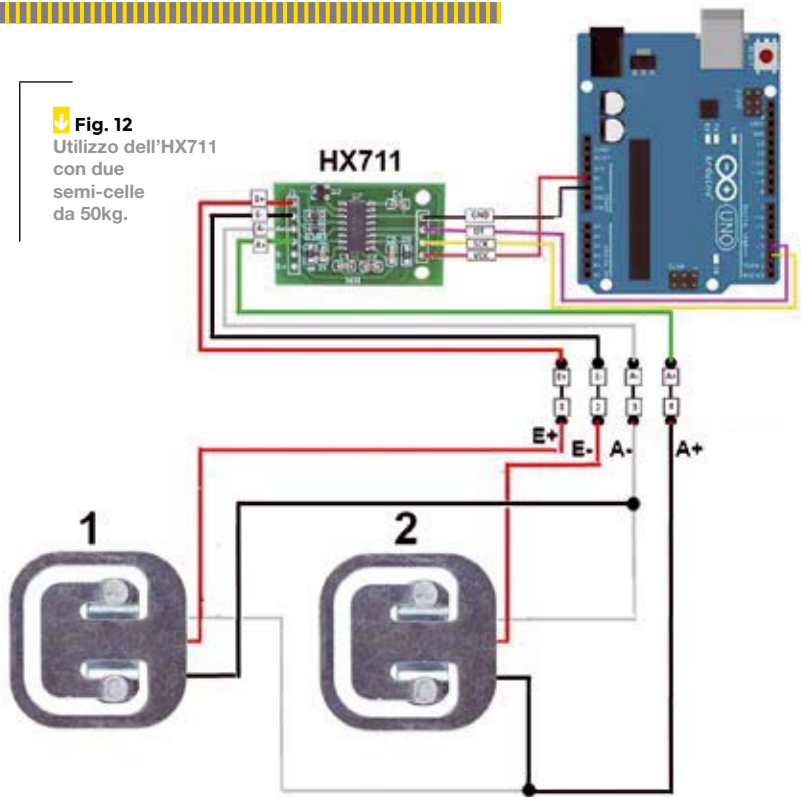


Fig. 13
Quattro semicelle di carico montate agli angoli di un supporto.

Listato 1

```
#include "HX711.h"
#define calibration_factor -9580.00 //This value is obtained using
//the SparkFun_HX711_Calibration sketch

#define DOUT 4
#define CLK 5
HX711 scale(DOUT, CLK);
void setup() {
  Serial.begin(9600);
  Serial.println("HX711 scale demo");
  scale.set_scale(calibration_factor);
  scale.tare(); //Assuming there is no weight on the scale
              //at start up, reset the scale to 0
  Serial.println("Readings:");
}
void loop() {
  Serial.print("Reading: ");
  float libbre = scale.get_units();
  float kg = (libbre/2.2046);
  Serial.print(kg, 1); //scale.get_units() returns a float
  Serial.print(" kg"); //You can change this to kg but you'll need
                      //to refactor the calibration_factor
  Serial.println();
}
```



```
#include "HX711.h"
#define calibration_factor -9580.00 //Fattore di calibrazione, ottenuto
//utilizzando lo sketch SparkFun_HX711_Calibration

#define DOUT 4 //Pin digitale DT
#define CLK 5 //Pin digitale SCK
HX711 scale(DOUT, CLK);

void setup() {
  Serial.begin(9600);
  Serial.println("Rilevazione Peso con Tara iniziale");
  scale.set_scale(calibration_factor);
  scale.tare(); //Assicurarsi che non ci sia nessun peso allo start.
  delay(500);
  Serial.println("Tara in corso... non mettere nessun peso!");
  delay(1000);
  Serial.println("Sistema pronto.");
  delay(300);
}

void loop() {
  Serial.print("Peso: ");
  float libbre = scale.get_units();
  float kg = (libbre/2.2046); //Conversione peso Libbre in Kg
  Serial.print(kg, 1);
  Serial.print(" kg");
  Serial.println();
  delay(500); //Ritardo nella lettura del peso. Diminuire per una lettura più frequente!
}
```

Queste celle di carico hanno le caratteristiche riportate di seguito:

- segnale di uscita = 1 mV ($\pm 0,1$);
- non linearità = 0,05(% sul fondo-scala);
- isteresi = 0,05 (% sul fondo-scala);
- ripetibilità = 0,05 (% sul fondo-scala);
- creep dopo 1 minuto = 0,1 (% sul fondo-scala);
- effetto della temperatura sull'uscita = 10°C (0,1% sul fondo-scala);
- effetto della temperatura sullo zero = 10°C (0,3% sul fondo-scala);
- resistenza d'ingresso = 1 kohm (± 10);
- resistenza di uscita = 1 kohm (± 10);
- resistenza di isolamento >2.000 Mohm
- temperatura di funzionamento = da -10°C a $+40^{\circ}\text{C}$;
- tensione di eccitazione ponte = 10 V;
- dimensioni (mm): 34x34x8.

Per le connessioni ad Arduino, ricordate che il filo rosso è il positivo (VCC) il nero il negativo (GND) e il bianco l'uscita (OUT) da collegare come mostrato. Il codice Arduino per eseguire la misura del peso con determinazione iniziale della tara lo trovate nel **Listato 2** e include la libreria HX711.

Fig. 14
Le quattro
semicelle collegate
a ponte.

+: White
-: Red
C: Black

Upper left Upper right Lower left Lower right

Dalla configurazione precedente, abbiamo ricavato un ulteriore sistema composto però da 4 mezze celle di carico dello stesso tipo, che potrebbe essere utilizzato per rilevare il peso di una cassa appoggiata sulle semicelle, disposte ognuna su un angolo di una struttura come quella proposta nella **Fig. 13**. Trattandosi di celle di carico (mezze

⬇ Listato 3

```
#include "HX711.h"
#define DOUT 4
#define CLK 5
const int buttonPin = 2;
int buttonState = 0;
HX711 scale(DOUT, CLK);
float calibration_factor = -9640.00; //-7050 worked for my 440lb max scale setup

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 calibration sketch");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");
  scale.set_scale();
  scale.tare(); //Reset the scale to 0
  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale. Useful in permanent scale projects.
  Serial.println(zero_factor);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    scale.set_scale();
    scale.tare(); //Reset the scale to 0
    long zero_factor = scale.read_average(); //Get a baseline reading
    Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale.
    Serial.println(zero_factor);
    pinMode(buttonPin, INPUT);
  }
  scale.set_scale(calibration_factor); //Adjust to this calibration factor
  Serial.print("Reading: ");
  float libbre = scale.get_units();
  float kg = (libbre/2.2046);
  Serial.print(kg, 1); //scale.get_units() returns a float
  Serial.print(" kg");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();
  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 10;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 10;
  }
}
```

celle) ciascuna capace di pesare 50 kg, possiamo arrivare a pesare un massimo di 200 kg in modo perfettamente bilanciato, giacché abbiamo gli elementi sui quattro angoli. Il circuito da realizzare con Arduino è descritto dallo schema di cablaggio proposto nella **Fig. 14**; da ogni singola cella viene "prelevato" il cavo rosso e collegato ai pin della scheda HX711. Questa configurazione a ponte di Wheatstone permette di avere un sistema bilanciato in cui il peso è misurato in modo equilibrato. Il codice per Arduino relativo a quest'ultimo esempio applicativo è proposto nel **Listato 3**. 



Cosa occorre?

I componenti utilizzati in questa presentazione sono disponibili presso Futura Elettronica. Arduino UNO REV3 (cod. ARDUINO UNO REV3) è venduto a Euro 24,50, la shield con LCD e INA125 (cod. WEIGHT_SHIELD) è disponibile a Euro 23,00, mentre la breakout con HX711 (cod. HX711BOARD) costa Euro 5,00. Un elenco completo delle celle di carico disponibili si trova all'indirizzo <http://bit.ly/loadcells>. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

GESTISCI E CONTROLLA LA TUA CASA CON I TELECONTROLLI GSM

ATTIVA E CONTROLLA DA REMOTO QUALSIASI DISPOSITIVO
ELETTRICO O ELETTRONICO: DALLA CENTRALINA ANTIFURTO
ALL'IMPIANTO DI IRRIGAZIONE, DALLE LUCI DI CASA
AL SISTEMA DI CONDIZIONAMENTO E RAFFRESCAMENTO.
GESTIONE TRAMITE CHIAMATA VOCALE O MESSAGGI SMS.

cod. TDG133

€ 79,00

Telecontrollo GSM bidirezionale 2 IN / 2 OUT

- 2 uscite a relè con funzionamento monostabile o bistabile e 2 ingressi d'allarme optoisolati.
- Dimensioni 103 x 67 x 28 mm
- Alimentazione 9÷32 Vdc/1 A
- Completo di antenna GSM esterna



**Disponibili
anche moduli
con funzione di:**

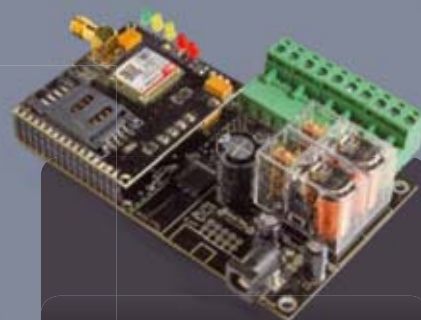
APRICANCELLO

cod. TDG134 • € 75,00

GSM

TERMOSTATO CON CONTROLLO GSM

cod. TDG139 • € 79,00



cod. TDG140 • € 88,00

TELECONTROLLO GSM CON COMANDI DTMF

Tutti i telecontrolli vengono forniti già montati e collaudati e sono facilmente configurabili mediante SMS o in locale tramite PC (per il collegamento al PC è necessario il modulo opzionale d'interfaccia USB cod. FT782M • € 13,00).



cod. TDG138 • € 79,00

TELEALLARME GSM CON ANTI JAMMER

Contenitore in ABS appositamente forato per ospitare i telecontrolli GSM: TDG133, TDG138 e TDG140. Cod. BOXTDG • € 16,90

Disponibile anche per TDG134. Cod. BOXTDG134 • € 16,90

Disponibile anche per TDG139. Cod. BOXTDG139 • € 16,90



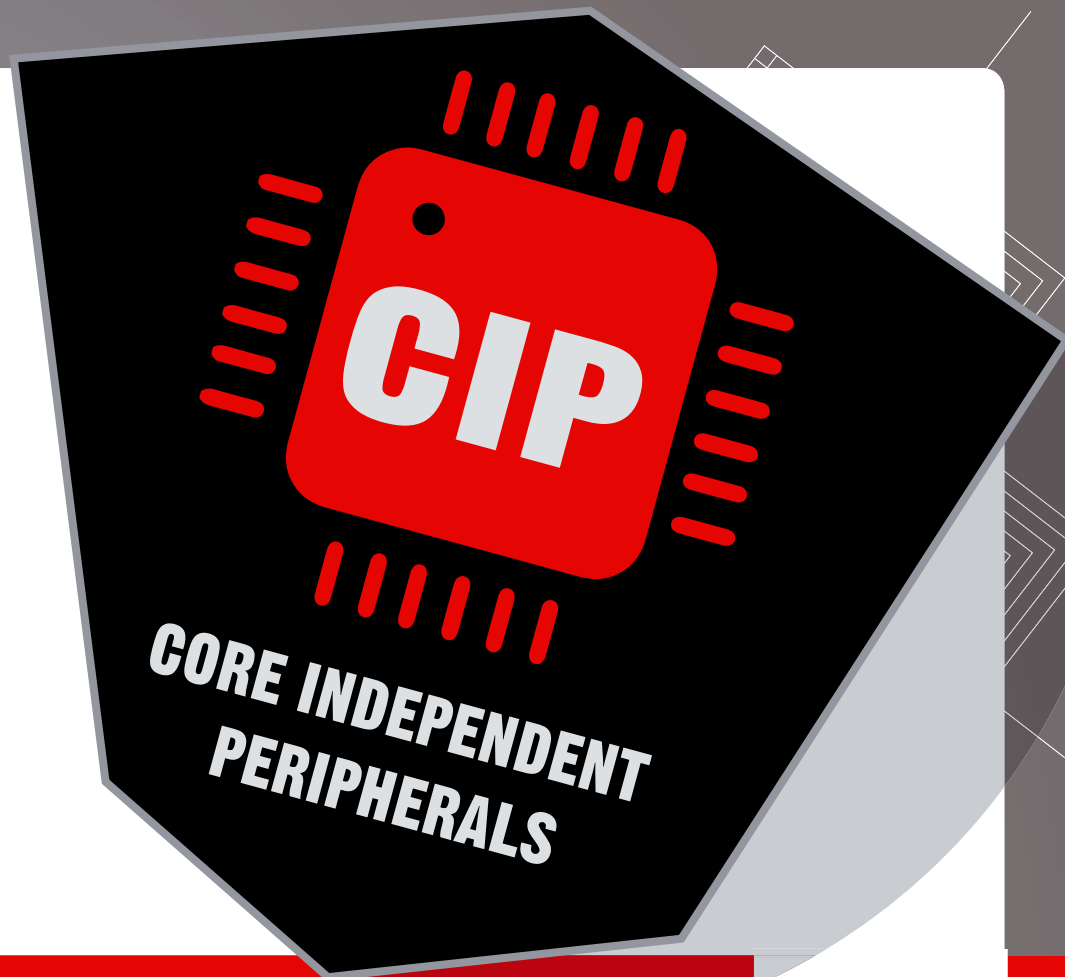
Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

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

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it



Conoscere e usare le **CORE INDEPENDENT PERIPHERALS**

di GUGLIELMO
BRAGUGLIA

Concludiamo la nostra carrellata sulle CIP, studiando quella che si trova solo sui microcontrollori di fascia più alta, siano esse a 8, 16 o 32 bit, ovvero quella relativa al DMA. Settima e ultima puntata.

Eccoci arrivati all'ultima parte di questo corso sulle CIP: in questa sede prenderemo in esame la Core Independent Peripheral che si trova solo sulle MCU di fascia più alta (siano esse a 8, 16 o 32 bit) ovvero quella relativa al DMA. A differenza di quanto fatto nelle precedenti puntate, pur rimanendo su una MCU a 8 bit dovremo utilizzare un hardware diverso e in particolare:

- la scheda di sviluppo "Curiosity HPC" (Part Number Microchip: DM164136) sostituendo la MCU di serie con una PIC18LF46K42;
- il modulo "IR sensor clic" di Mikroe (Mikroe M-2677);
- il "TinyTerminal" presentato sul n° 235 di Elettronica In;
- l'ambiente di sviluppo MPLAB X, che è alla versione 5.15

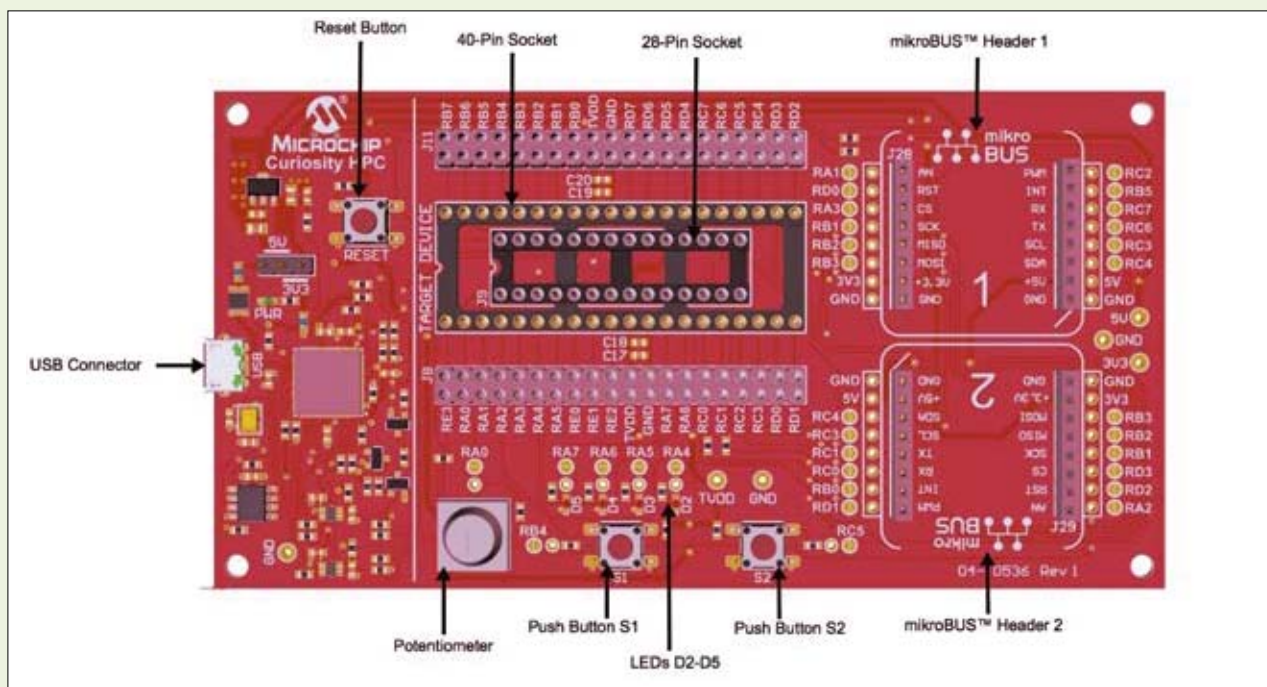


Fig. 1 - Scheda Microchip "Curiosity HPC" con indicazione delle periferiche.

- il plugin MCC (MPLAB® Code Configurator), che è alla versione 3.75;
- il compilatore XC8, attualmente alla versione 2.05.

La scelta della "Curiosity HPC" si è resa obbligatoria per mostrare il funzionamento della CIP DMA, dato che essa è presente solo su MCU ad 8 bit di fascia alta (oltre che, naturalmente, su MCU a 16 e 32 bit) con "case" normalmente di dimensioni superiori al massimo montabile sulla "Curiosity" standard che abbiamo usato nelle precedenti puntate.

La MCU scelta per dimostrare la CIP DMA è difatti il PIC18LF46K42, che è disponibile in "case" PDIP40 e che può essere montato sulla suddetta scheda di sviluppo (**Fig. 1**). Detta scheda offre inoltre la possibilità di inserire sino a due moduli aggiuntivi formato "mikroBUS" ed è dotata, come sempre, di alcuni LED (4 LED indicati come D2, D3, D4 e D5), di due pulsanti (S1 ed S2), di un potenziometro e dell'immancabile programmatore/debugger che permette la programmazione ed il test dell'applicazione sviluppata, senza la necessità di alcun hardware aggiuntivo.

A scopo dimostrativo si è utilizzato poi, come fonte dei dati da visualizzare, il modulo Mikroe "IR Sensor Clic", su cui è presente un chip AK9750 con quattro sensori IR (disposti a croce) e con interfaccia I²C. L'area di "sensibilità" di ciascun sensore IR è disegnata sul modulo ed è ben visibile che le quattro aree si sovrappongono lungo le diagonali permettendo una copertura a 360°. Ogni sensore genera un segnale in corrente che viene convertito all'interno del chip, tramite un convertitore a 16 bit, in una tensione che è possibile leggere numericamente attraverso l'interfaccia I²C. Nel chip è anche

presente un sensore interno di "temperatura", anche esso sempre leggibile attraverso la stessa interfaccia. Per la visualizzazione dei dati letti si è scelto di utilizzare il "TinyTerminal": si tratta di un piccolo e semplice terminale ASCII, visibile nella **Fig. 3**, in grado di visualizzare su un display OLED, da 42 caratteri per 8 righe, i dati ricevuti alla velocità di 9.600 bps sulla porta seriale. Tramite una serie di caratteri di controllo, è anche possibile effettuare il posizionamento del cursore o assegnare particolari "attributi" video ai caratteri stampati.

Dato che il programma dimostrativo dovrà evidenziare il vantaggio di usare la CIP DMA, piuttosto che la scrittura diretta da parte della CPU sulla porta seriale, non si farà uso di tali caratteri di posizionamento, ma anzi, si cercherà di trasmettere sempre il maggior numero di caratteri possibile, proprio

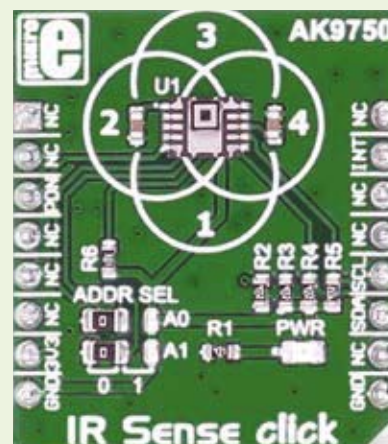


Fig. 2 - Scheda Mikroe "IR Sensor Clic".

affinché la differenza di tempo tra l'utilizzo ed il non utilizzo del DMA sia apprezzabile ad occhio nudo.

In pratica, compito del programma dimostrativo, sarà:

- far lampeggiare, seguendo un conteggio binario, con incremento ogni 50 ms, i quattro LED presenti sulla scheda;
- leggere l'eventuale pressione del pulsante "S2" e commutare lo stato di un flip/flop, leggibile da programma, per indicare se si vuole la trasmissione dei dati via seriale a carico della CPU o del DMA;
- leggere dal bus I²C, ogni 750 ms., il valore dei quattro sensori IR e quello del sensore di temperatura e memorizzare i valori;
- stampare sulla seriale, collegata al "TinyTerminal", i valori letti, impegnando la CPU o il DMA in funzione della scelta dell'utilizzatore (scelta effettuata con il pulsante "S2");

Dato che si cercherà, ai fini didattici, di massimizzare il numero di caratteri trasmessi via seriale, sarà ben visibile la differenza fra la trasmissione a carico della CPU e quella effettuata tramite DMA: nel primo caso, dato il tempo impiegato per trasmettere i caratteri, si noterà un avanzamento "a scatti" del conteggio binario sui LED (infatti la trasmissione seriale effettuata dalla CPU impiegherà più dei 50 ms con cui è richiesto l'avanzamento del contatore), mentre nell'utilizzo del DMA il conteggio continuerà a scorrere in modo fluido poichè la CPU sarà sgravata dalla trasmissione seriale.

DMA (DIRECT MEMORY ACCESS)

Come accennato, la CIP di cui ci occuperemo è il DMA, ovvero quella CIP che ha il compito di "spostare" dati all'interno della MCU, senza richiedere, durante lo "spostamento", alcun intervento da parte della CPU. Lo schema a blocchi di funzionamento di tale CIP è visibile in **Fig. 4**.

La **Tabella 1** riassume i possibili tipi di spostamento: vediamo che mentre la memoria programma e la EEPROM possono essere utilizzate solo come origine dei dati, la memoria dati (GPR) e tutti i registri e le porte di I/O (SFR) possono essere utilizzati sia come origine che come destinazione.

In aggiunta a ciò, una larga parte dei segnali disponibili dalle periferiche (ad esempio segnali interrupt), possono essere

ORIGINE DEI DATI	DESTINAZIONE DEI DATI
Memoria "Flash" dove è il programma	GPR (General Purpose Registers)
Memoria "Flash" dove è il programma	SFR (Special Function Registers)
Memoria EEPROM	GPR (General Purpose Registers)
Memoria EEPROM	SFR (Special Function Registers)
GPR (General Purpose Registers)	GPR (General Purpose Registers)
GPR (General Purpose Registers)	SFR (Special Function Registers)
SFR (Special Function Registers)	GPR (General Purpose Registers)
SFR (Special Function Registers)	SFR (Special Function Registers)

Tab. 1 - Spostamenti dei dati.

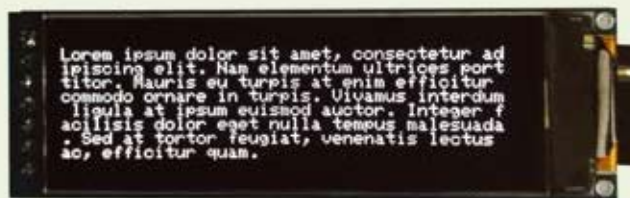


Fig. 3 - Il display del "TinyTerminal".

utilizzati per avviare o arrestare il trasferimento dei dati in modo automatico e senza l'intervento della CPU. È anche possibile iniziare ed arrestare in qualsiasi momento detto trasferimento da software. Prima di procedere con l'esame dell'applicazione, esaminiamo più in dettaglio il funzionamento della CIP DMA. Il movimento di dati attraverso il DMA avviene per "Transazioni" che vengono svolte in due passi; in un primo passo il dato di "origine" viene spostato in un buffer di appoggio, in un secondo passo, questo buffer di appoggio viene spostato nella "Destinazione". Nel nostro caso il buffer

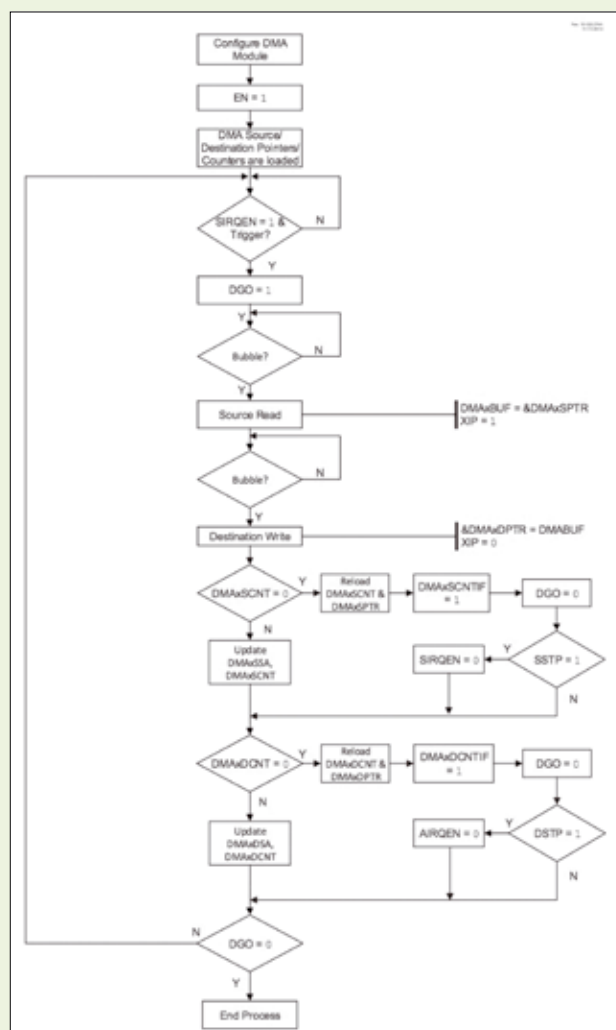


Fig. 4 - Diagramma di flusso del modulo DMA.

DIMENSIONE ORIGINE	DIMENSIONE DESTINAZIONE
1 byte	N bytes
2 bytes	2 x N bytes
2 x N bytes	2 bytes
N bytes	1 byte

Tab. 2 - Dimensioni di origine e destinazione dei messaggi.

di appoggio è un registro a 16 bit quindi la singola "Transazione" può spostare uno o due byte.

Per spostare un "Messaggio" (ovvero lo spostamento dell'intero blocco di dati) vengono effettuate una o più "Transazioni" in funzione delle dimensioni della quantità di byte da spostare. Le dimensioni dell'origine e della destinazione possono essere differenti purché multipli l'uno dell'altro.

La **Tabella 2** riassume le possibili dimensioni dell'origine e della destinazione per ciascun Messaggio.

Sia l'Origine che la Destinazione sono identificate da due indirizzi che possono rappresentare una locazione di memoria programma (solo Origine), una locazione di EEPROM (solo Origine) oppure una locazione di memoria dati o un registro. Oltre a specificare l'indirizzo dell'Origine e della Destinazione, occorre specificare cosa occorre fare su tali indirizzi per ogni Transazione. Le possibilità sono: fare nulla, incrementare l'indirizzo, decrementare l'indirizzo. Ad esempio, se si vuole trasferire un blocco di dati dalla SRAM alla porta seriale (UART), occorrerà specificare che ad ogni Transazione l'indirizzo di origine va incrementato (sino al massimo della lunghezza del

blocco da trasferire), mentre l'indirizzo di destinazione deve rimanere costante (sarà quello del registro TX della UART). Infine occorrerà specificare gli "automatismi" che dovranno essere applicati, ovvero quando incrementare o decrementare un indirizzo, quando considerare concluso il trasferimento e cosa fare in caso di errore.

Per fare quanto sopra descritto, la MCU dispone di una serie di registri associati a ogni canale di DMA: ad esempio il PIC18LF46K42 ha due canali di DMA e quindi due serie di registri identici, una per canale.

Volendo approfondire, tutti i dettagli operativi e i registri utilizzati sono esaminabili nel datasheet del PIC18LF46K42 (DS40001919B), sezione 15 e nel documento Microchip "Direct Memory Access on 8-bit PIC Microcontrollers" (DS90003164B). Per la nostra applicazione, come al solito, utilizzeremo MCC, che ci permette di lavorare senza dover andare ad esaminare esattamente dove e quali bit occorre impostare, evitandoci possibili errori di configurazione e generando il codice 'C' per noi, codice che poi adatteremo per il nostro caso particolare.

ESEMPIO APPLICATIVO CON DMA

Nell'applicazione d'esempio che andremo a sviluppare, utilizzeremo, come già fatto in passato, varie CIP ed MCC per realizzare l'applicazione schematizzata nella **Fig. 5**. In pratica, un Timer (Timer6) scandirà i millisecondi e, tramite una ISR, farà avanzare un contatore che sfrutteremo per alcune temporizzazioni del nostro "super-loop" nel main. Un altro Timer (Timer4) assieme ad una CLC (CLC1) effettuerà il

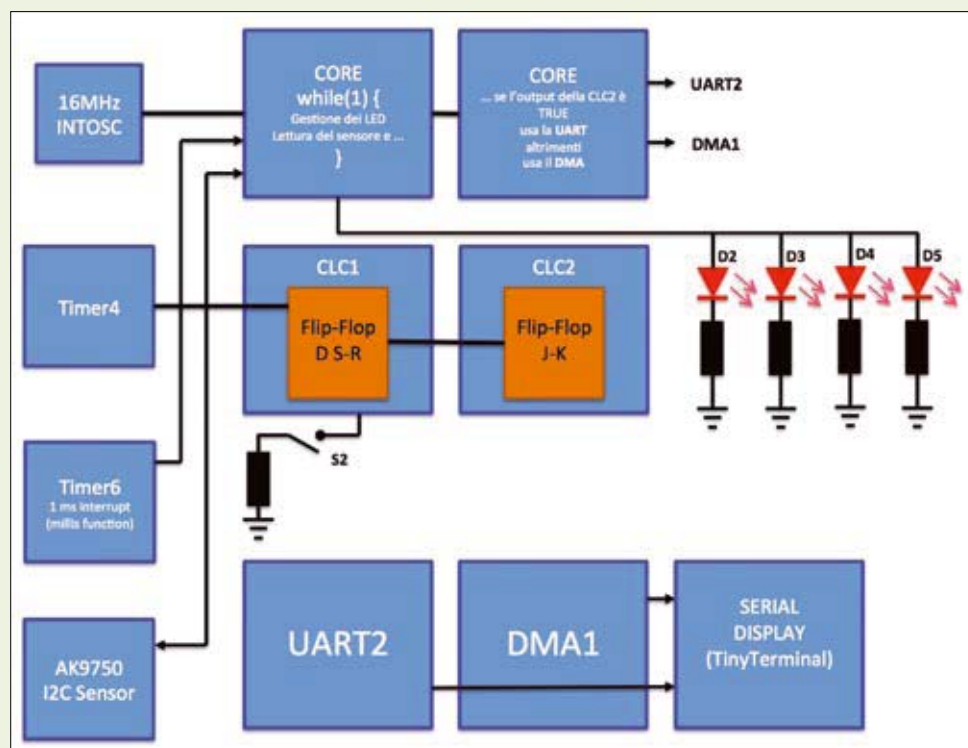


Fig. 5 - Schema a blocchi del progetto di esempio.

debouncing del pulsante S2 presente sulla "Curiosity HPC" e l'uscita di detta CLC entrerà in una ulteriore CLC (CLC2) utilizzata come flip-flop J-K (così da avere un'uscita che si alterna tra HIGH e LOW ad ogni pressione di S2). Ogni 50 ms si farà avanzare un contatore binario a 4 bit il cui valore sarà rappresentato sui 4 LED della scheda (si vedrà un conteggio binario avanzare da 0X00 a 0X0F e poi ricominciare), mentre, ogni 750 ms si effettuerà la lettura dei sensori del AK9750 e, in funzione del valore della CLC2 (HIGH o LOW) si visualizzeranno i valori su un terminale seriale, a 9.600 baud, scrivendo sulla UART o via software (tramite la funzione `printf()`) o via DMA. Nel caso di scrittura via software si potrà osservare che, dato il tempo impiegato per trasmettere tutti i caratteri sulla seriale, l'avanzamento ogni 50 ms del contatore visualizzato sui LED, sarà irregolare (ci sarà un arresto ogni volta che si andrà a fare la `printf()`). Nel caso di scrittura via DMA, avvenendo essa indipendentemente dalla CPU, si potrà osservare che il conteggio procederà spedito senza interruzioni anche durante la scrittura sulla seriale.

Come ormai abbiamo imparato a fare, metteremo in esecuzione MPLAB X e creeremo un nuovo progetto di tipo standalone; come MCU selezioneremo il PIC18LF46K42, MCU che avremo provveduto a montare sulla Curiosity HPC (Microchip P/N# DM164136) e come compilatore XC8 v2.05. Andremo poi sulle proprietà del compilatore (XC8 Global Options) e, nella categoria **Global Options**, alla voce **C standard** selezioneremo **C90** (rammentate che se non si effettua questa selezione e si lascia il default C99, si potranno verificare errori durante la compilazione del codice).

A questo punto potremo mettere in esecuzione il tool MCC per la solita configurazione iniziale. Selezioneremo come "Oscillator Select" HFINTOSC, come "HF Internal Clock" 16 MHz; e come "Clock Divider" il valore 1 e ci accerteremo che il "Watchdog" sia disabilitato. Nella finestra in basso, tramite

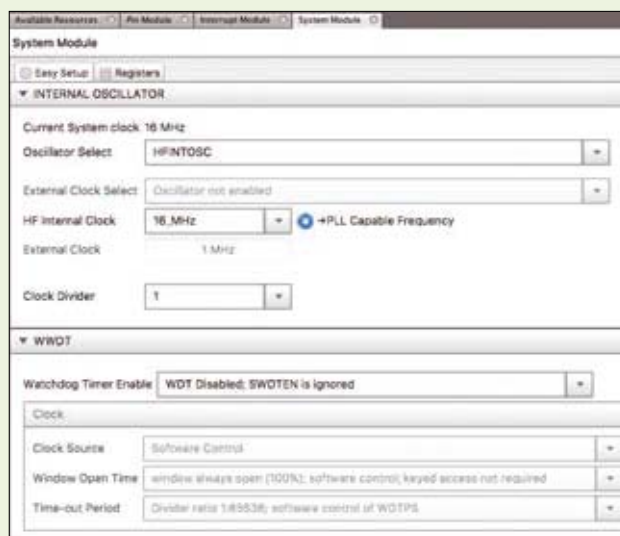


Fig. 6 - Maschera di MCC per la configurazione del sistema.

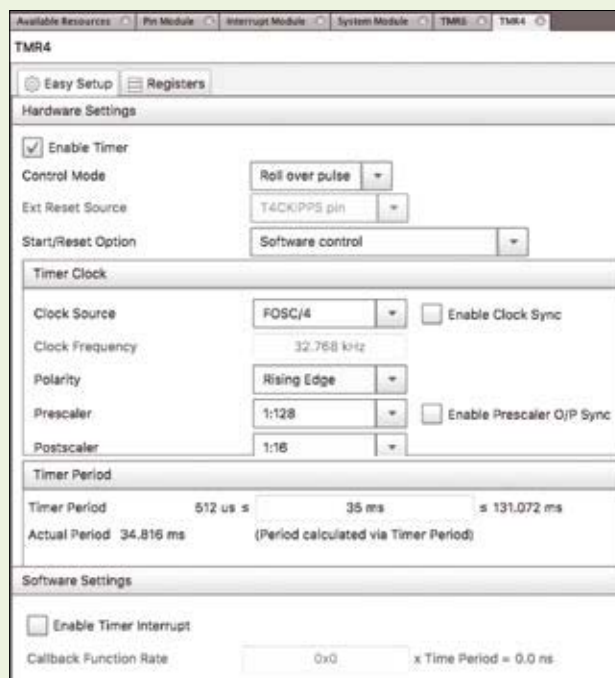


Fig. 7 - Maschera di MCC per la configurazione del Timer6.

il menu a tendina, ci dovremo ricordare di selezionare come "Package" il PDIP40 dato che, in tale case, è il chip che avremo montato sulla "Curiosity HPC".

Terminata la configurazione iniziale andremo a configurare il "Timer" che ci permetterà di generare l'interrupt ogni 1 ms, interrupt che farà poi avanzare un contatore che utilizzeremo per schedare nel tempo alcune attività nel main (nel super-loop). Sulla sinistra, selezionato il tab "Resource management (MCC)", e nella sezione "Devices Resources" apriremo, facendoci sopra un clic, la risorsa "Timer".

Come già visto nelle altre occasioni, ci verranno presentati i vari "Timers" presenti nella MCU e noi selezioneremo il "Timer6". Facendo doppio clic su di esso, al centro si aprirà un nuovo tab relativo a detto "Timer6", dove potremo procedere alla sua configurazione. Verificata, come sempre, la presenza del segno di spunta sulla casella "Enable Timer", selezioneremo come "Clock Source" FOSC/4, come "Postscaler" il valore 1/1 e come "Prescaler" il valore 1/32, che ci permetterà di selezionare il periodo del Timer da un minimo di 8 µs a un massimo di circa 2.048 ms. Sceglieremo come periodo 1 ms, valore che ci darà un periodo reale esattamente di 1 ms, senza arrotondamenti. Metteremo, infine, il segno di spunta nel campo "Enable Timer Interrupt" assegnando al campo "Callback Function Rate" il valore 0x1, valore per cui, ad ogni interrupt verificatosi, verrà chiamata la funzione di "Call Back" che MCC predisporrà nel sorgente (e che dovremo completare per la gestione del contatore dei millisecondi).

Terminata la configurazione del Timer che genererà la nostra "base dei tempi", passeremo a configurare il Timer che ci occuperà per il debouncing del pulsante S2, così come abbiamo



Fig. 8 - Maschera di MCC per la configurazione del Timer4.

già imparato a fare nella seconda puntata di questo corso. Sulla sinistra, selezionato di nuovo il tab "Resource mana-

gement (MCC)", apriremo, facendoci sopra un clic, sempre la risorsa "Timer". Ci verranno presentati i vari "Timers" presenti nella MCU e qui selezioneremo il "Timer4". Facendo doppio clic su di esso, al centro dello schermo si aprirà un nuovo tab relativo a detto "Timer4", dove potremo procedere alla configurazione corrispondente (**Fig. 8**). Verificata, come sempre, la presenza del segno di spunta sulla casella "Enable Timer", selezioneremo come "Clock Source" FOSC/4, come "Postscaler" il valore 1/16 e come "Prescaler" il valore 1/128 che ci permetterà di selezionare il periodo del Timer da un minimo di 512 μ sec ad un massimo di circa 131.072 ms. Scegliere-

mo come periodo di debouncing 35 ms, valore che ci darà un periodo reale esattamente di 34.816 ms. Passeremo ora ad inserire la CIP "CLC" che utilizzeremo per il debouncing vero e proprio. In "Resource management (MCC)", nella sezione "Devices Resources", apriremo, facendoci sopra un clic, la risorsa "CLC". Ci verranno presentati le varie "CLC" presenti nella MCU; selezioneremo la "CLC1" facendo doppio clic su di essa. Al centro, si aprirà un nuovo tab, relativo a detta "CLC1" dove potremo procedere alla sua configurazione (**Fig. 9**).

Ci accerteremo, come d'abitudine, della presenza del segno di spunta sulla casella "Enable CLC" e verificheremo che non ci siano altri segni di spunta nelle altre caselle.

Nel menu a tendina chiamato "Mode" selezioneremo "1-input D flip-flop with S and R" e, nel menu a tendina, collegato

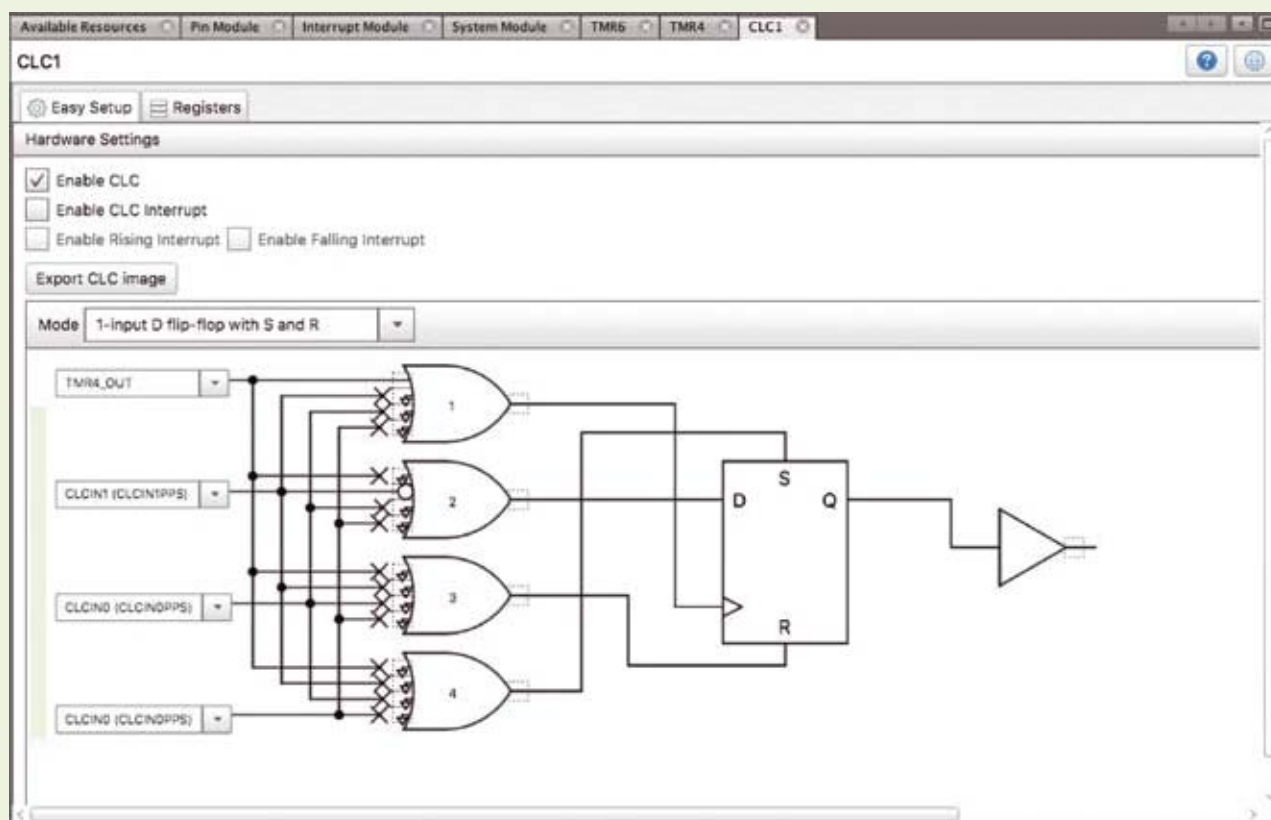


Fig. 9 - Maschera di MCC per la configurazione della CLC1.

alla prima porta logica di "Ingresso", selezioneremo TMR4_OUT ovvero il segnale in uscita dal Timer4. Faremo quindi un solo clic nel rettangolino tratteggiato corrispondente così da collegare tale segnale al primo pin della prima porta di "Ingresso".

Nel secondo menu a tendina, collegato alla seconda porta logica di "Ingresso", selezioneremo CLCIN1 (CLCIN1PPS), faremo un primo clic nel rettangolino tratteggiato corrispondente, così da collegare tale segnale al secondo pin della seconda porta di "Ingresso" ed un secondo clic per far apparire il 'o' indicante la negazione del segnale. Notare che CLCIN1 è connesso al pin "C5", pin corrispondente a quello su cui è collegato il pulsante "S2".

Dato che a noi occorre un segnale che rimanga stabile, alternandosi tra HIGH e LOW, ad ogni pressione del pulsante "S2" faremo seguire alla CLC appena configurata una ulteriore CLC che svolgerà le funzioni di un flip-flop di tipo J-K. Tornati nel "Resource management (MCC)", nella sezione "Devices Resources", apriremo, facendoci sopra un clic, la risorsa "CLC". Come sempre ci verranno presentati le varie "CLC" presenti nella MCU; selezioneremo la "CLC2" facendo doppio clic su di essa. Al centro, si aprirà un nuovo tab, relativo a detta "CLC2" dove potremo procedere alla sua configurazione (**Fig. 10**). Verificato il solito segno di spunta sulla casella "Enable CLC" e verificato che non ci siano altri segni di spunta nelle altre caselle, nel menu a tendina chiamato "Mode" selezioneremo

"JK flip-flop with R" e, nel menu a tendina, collegato alla prima porta logica di "Ingresso", selezioneremo CLC1_OUT ovvero il segnale in uscita dalla CLC1 e faremo un solo clic nel rettangolino tratteggiato corrispondente così da collegare tale segnale al primo pin della prima porta di "Ingresso". Provvederemo infine a fare un clic nei due rettangolini tratteggiati relativi all'uscita della seconda e quarta porta di "Ingresso" (corrispondenti ai segnali J e K del flip-flop) in modo che appaia 'o' indicante la "negazione" del segnale in uscita. In questo modo avremo configurato la nostra CLC così che, ogni volta che arriva un impulso dalla CLC1 (pressione del nostro pulsante S2), venga inviato un segnale sul clock del flip-flop e che esso commuti il segnale in uscita tra HIGH e LOW, segnale che utilizzeremo da software per scegliere se inviare i dati tramite la funzione printf() o tramite DMA.

Per completare la nostra applicazione, dobbiamo ancora configurare tre CIP: la CIP DMA, quella che gestisce il bus I²C-Bus e la UART. Torneremo quindi nel "Resource management (MCC)" e nella sezione "Devices Resources", apriremo, facendoci sopra un clic, la risorsa "DMA". Ci verranno presentati i vari "DMA" presenti nella MCU e noi selezioneremo il "DMA1" facendo doppio clic su di esso. Al centro, si aprirà un nuovo tab, relativo a detto "DMA1" dove potremo procedere alla sua configurazione. Dato il numero di voci da configurare si è diviso il tutto in tre schermate dove sono presenti tutti gli elementi che occorre programmare.

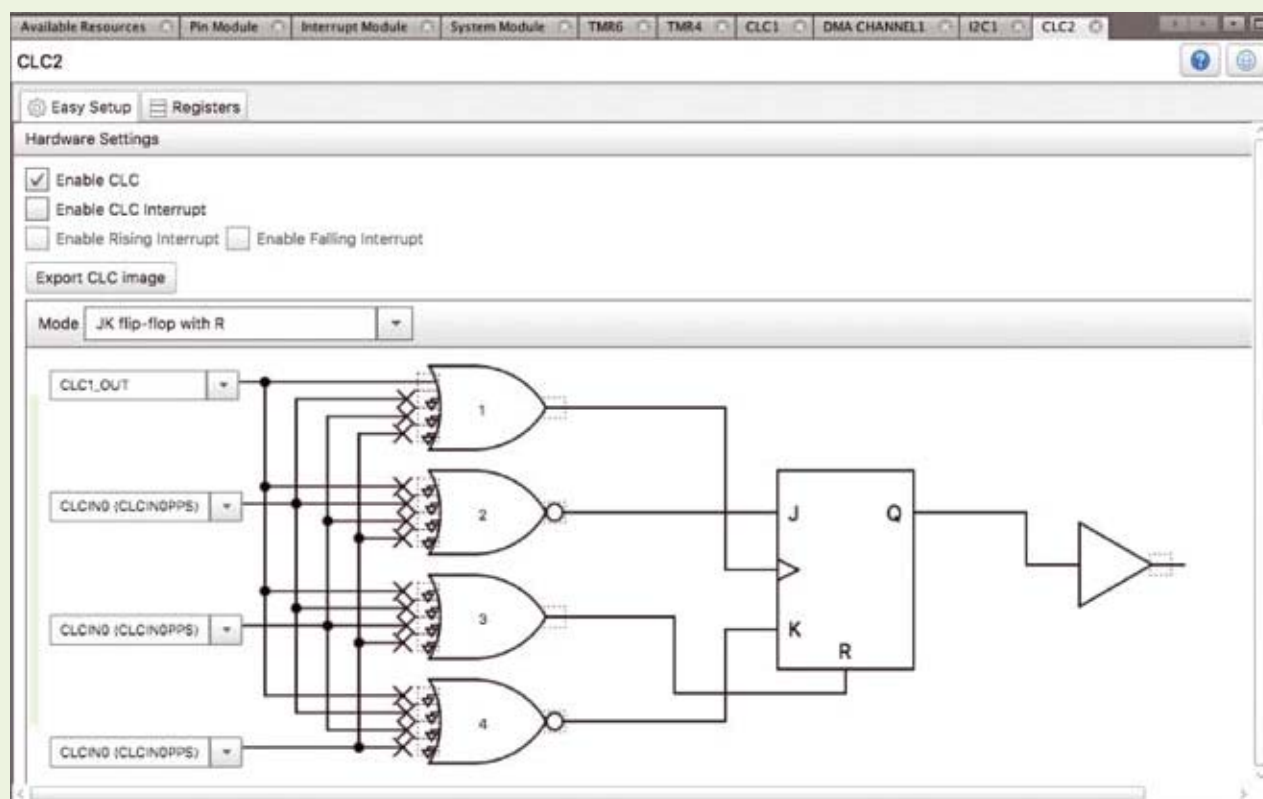


Fig. 10 - Maschera di MCC per la configurazione della CLC2.

Nella prima parte del tab dobbiamo prima di tutto verificare che non ci sia alcun segno di spunta relativamente agli interrupt (in questa applicazione non li utilizziamo), dopo di che passiamo a configurare:

- la configurazione di base;
- le informazioni in merito alla destinazione dei dati;
- le informazioni in merito a dove prendere i dati;
- le informazioni sulla dimensione dei dati in ingresso ed in uscita;
- le informazioni sulla disponibilità dei dati e come muoversi nell'origine e nella destinazione.

Questo lo vedete nella **Fig. 11**, che propone la parte superiore della schermata (il tab non entra tutto in uno screenshot). Dato che non utilizziamo il "Abort of Transfer Interrupt" non ci interessa programmare il valore del campo "AIRQ" che lasciamo di default a LVD e nel menu a tendina di nome "AIRQEN" selezioniamo disabled.

Lasciamo sempre al valore di default not in progress il campo

"DGO" e selezioniamo per il campo "EN" il valore di disabled (abileremo il DMA da programma dopo un'ulteriore configurazione a run-time).

Poiché, come vedremo in seguito, utilizzeremo lo "Start of Transfer Interrupt", imposteremo il campo "SIRQEN" a enabled e passeremo alla configurazione degli altri registri che troviamo nella figura successiva (**Fig. 12**).

Per il campo "DMODE", dato che la nostra destinazione è il registro TX della UART2, selezioneremo il valore unchanged (difatti non dobbiamo né incrementare né decrementare tale indirizzo), per "DSTP" selezioneremo not cleared (che indica "SIRQEN bit is not cleared when Destination Counter reloads"), per SMODE, dato che la nostra origine dei dati è un'area di memoria, dovremo incrementare l'indirizzo per ogni carattere trasmesso e quindi selezioneremo il valore incremented, indicheremo poi che l'origine dei dati è SFR/GPR (ovvero registri o dati in memoria RAM) e imposteremo SSTP a cleared (che indica "SIRQEN bit is cleared when Source Counter reloads").

Fig. 11 - Maschera di MCC per la configurazione del DMA1 (prima parte).

Fig. 12 - Maschera di MCC per la configurazione del DMA1 (seconda parte).

Dal datasheet verifichiamo l'indirizzo corrispondente al registro di trasmissione della UART2 (U2TXB) che risulta essere 0x3DD2 e quindi mettiamo la parte alta di tale indirizzo nel campo "DSA" e la parte bassa in "SSA", inoltre, essendo tale registro di un byte metteremo in "DSZ" del registro DMA1DSZH (parte alta) il valore 0x0 mentre in "DSZ" del registro DMA1DSZL (parte bassa) il valore 0x1 e passeremo a configurare gli altri registri riportati nella **Fig. 13**.

Nel campo "SIRQ" indicheremo che il DMA potrà passare ad elaborare la "transizione" successiva del "messaggio" da trasferire (nel nostro caso la transizione è di 1 byte) quando il registro U2TXB è vuoto e quindi selezioneremo appunto il valore U2TX.

Terminata la parte relativa alla destinazione dei dati ci sono i registri relativi all'origine dei dati, ma, trattandosi di area di memoria RAM (in pratica una variabile allocata nel programma) tali registri verranno impostati dall'applicativo e quindi in MCC lasciati a zero.

Come detto, il sensore utilizzato nell'applicazione è il AK9750

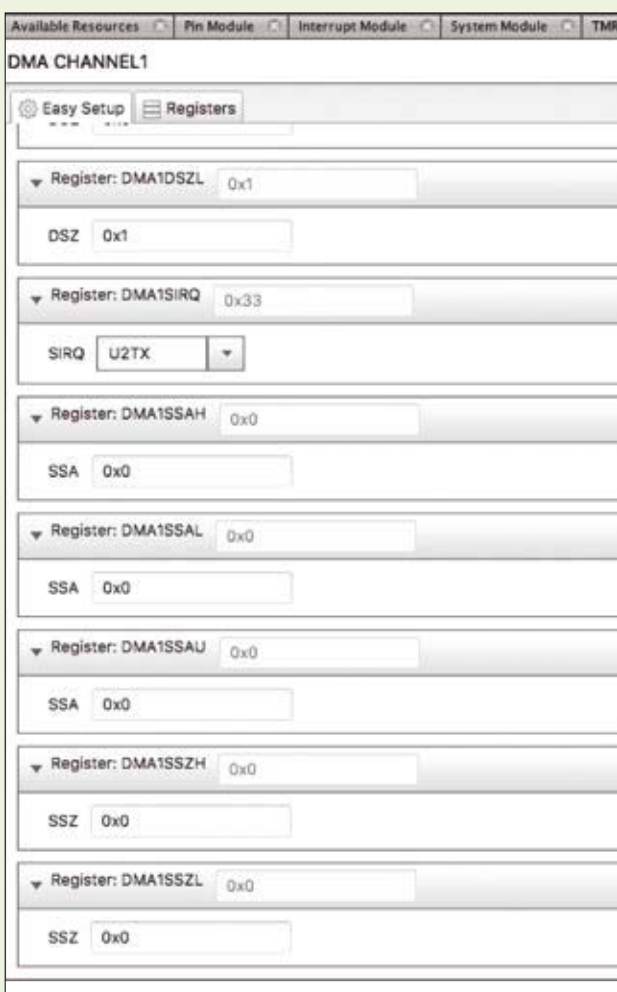


Fig. 13 - Maschera di MCC per la configurazione del DMA1 (terza parte).

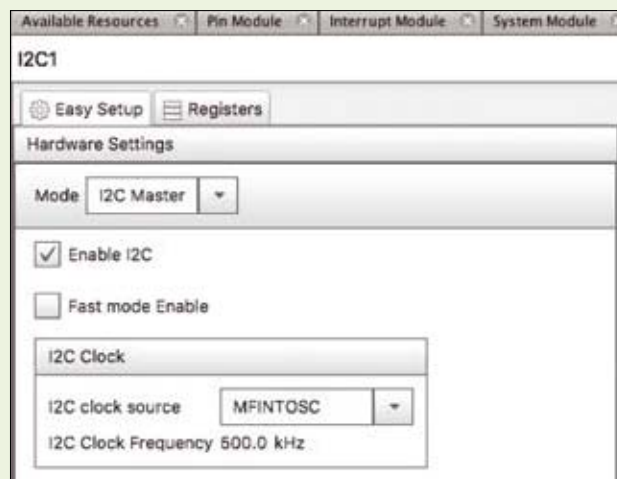


Fig. 14 - Maschera di MCC per la configurazione del I2C..

che si trova montato su una clic board di Mikroe. Tale sensore comunica in I2C e quindi, sempre con l'ausilio di MCC dobbiamo ora configurare la CIP che si occupa dei trasferimenti I2C. Sulla sinistra, selezionato il tab "Resource management (MCC)", nella sezione "Devices Resources" apriremo, facendoci sopra un clic, la risorsa "I2C".

Come già visto nelle altre occasioni, ci verranno presentati i vari moduli di interfaccia I2C-Bus presenti nella MCU e noi selezioneremo il "I2C1". Facendo doppio clic su di esso, al centro si aprirà un nuovo tab, relativo a detto "I2C1", dove potremo procedere alla sua configurazione (**Fig. 14**).

Verificata, come sempre, la presenza del segno di spunta sulla casella "Enable I2C", nel menu a tendina relativo al "Mode" selezioneremo I2C Master e quindi selezioneremo come "I2C clock source" MFINTOSC così da avere una frequenza di clock per il bus I2C di 500.0 KHz

Potremo così passare ora alla configurazione dell'ultima CIP che ci occorre, ovvero la UART. Di nuovo, selezionato il tab "Resource management (MCC)", nella sezione "Devices Resources", apriremo, facendoci sopra un clic, la risorsa "UART" dove selezioneremo, con il solito doppio clic, la risorsa UART2 (PIC10/PIC12/PIC16/PIC18 MCUs by Microchip Technology Inc.) che potremo passare a configurare (**Fig. 15**).

Per il campo "Mode" selezioneremo Asynchronous 8-bit mode, metteremo il segno di spunta ai campi "Enable UART" ed "Enable Transmit" (non la utilizzeremo in ricezione e quindi non attiveremo tale funzione); selezioneremo poi, per il campo "Baud Rate", il valore 9600, lasceremo gli altri campi al valore predefinito e termineremo mettendo il segno di spunta al campo "Redirect STDIO to UART".

L'ultima operazione che rimane da fare in MCC è la configurazione dei PIN di I/O tramite la griglia di configurazione che si trova a fondo pagina nel TAB "Pin Manager (Grid View)" e l'assegnazione di nomi logici ai pin dei LED tramite il TAB in alto di nome "Pin Module".

Nel TAB "Pin Manager (Grid View)" imposteremo come ou-

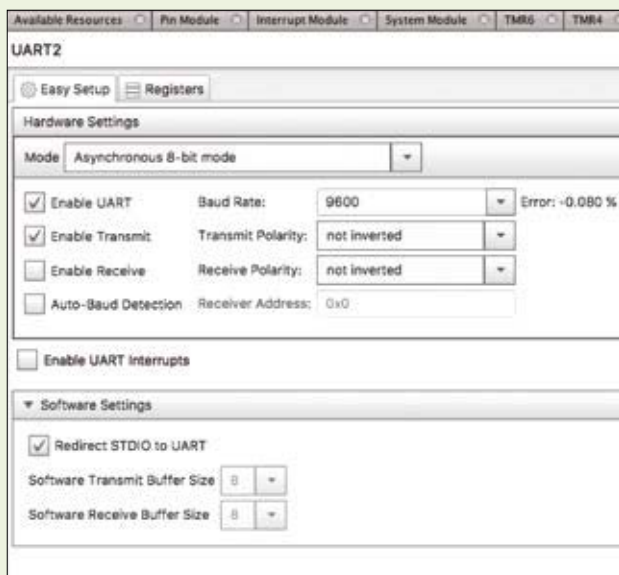


Fig. 15 - Maschera di MCC per la configurazione della UART2.

tput i GPIO relativi ai LED, ovvero A4, A5, A6 ed A7 ed il pin A3 utilizzato dal module Mikroe per il segnale di PDN del sensore AK9750 (**Fig. 16**); selezioneremo come pin di input il GPIO relativo al pin INT del suddetto modulo Mikroe, ovvero il pin B5; selezioneremo, se non già selezionati, come UART2 TX il pin B1 e come RX il pin B7 ed infine verificheremo che i pin del I2C1 siano correttamente impostati a C3 per il pin SCL e C4 per il pin SDA.

Nel TAB "Pin Module" (**Fig. 17**) assegneremo i nomi logici ad alcuni pin; in particolare, AK9750_PDN al pin RA3, AK9750_INT al pin B5 ed i nomi LED_D2, LED_D3, LED_D4 e LED_D5 ai pin A4, A5, A6 ed A7.

Con questo, la parte relativa alla configurazione con MCC è terminata ed effettuata la "Generazione" del codice, dovremo passare a scrivere il nostro applicativo, sia modificando alcuni file di MCC, sia aggiungendo alcuni moduli (il driver per il sensore AK9750) e modificando il file **main.c**.

I moduli software per il sensore AK9750, che trovate con il nome di **drvAK975x.c** e **drvAK975x.h**, scaricabili dal nostro sito www.elettronica.in.it assieme al progetto MPLAB X completo, vanno aggiunti nel progetto tra i files .h e tra i file.c. I primi due file che andremo a modificare sono quelli relativi al Timer6 per la gestione dell'interrupt e la creazione di due funzioni di utilità che verranno usate nel **main()**.

Una volta aperto il file "tmr6.c" dovremo aggiungere subito all'inizio, dopo la **#include "tmr6.h"**, le righe di codice che trovate nel **Listato 1**.

Ci sposteremo poi verso la fine del file, dove si trova la funzione "void TMR6_Callback(void)" che completeremo come mostrato nel **Listato 2**.

Terminata la modifica del file **tmr6.c** dovremo passare al suo associato **tmr6.h** dove, alla fine, prima della riga **#ifndef __cplusplus // Provide C++ Compatibility** inseriremo le seguenti righe:

```
/**
 * User defined functions
 */

uint32_t millis(void);
void delay(uint32_t);

/**
 * End of User defined functions
 */
```

Notifications			Output - MPLAB® Code Configurator								Notifications [MCC]								Pin Manager: Grid View																							
Package:	POIP40	▼	Pin No:		2	3	4	5	6	7	14	13	33	34	35	36	37	38	39	40	15	16	17	18	23	24	25	26	19	20	21	22	27	28	29	30	8	9	10	1		
			Port A ▼								Port B ▼								Port C ▼								Port D ▼								Port E ▼							
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3				
CLC1	CLC1	output	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌													🔌	🔌	🔌	🔌	🔌	🔌	🔌													
CLC2	CLC2	output																						🔌	🔌	🔌	🔌	🔌	🔌	🔌												
CLCx ▼	CLCIN0	input	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌														🔌	🔌	🔌	🔌	🔌	🔌	🔌												
	CLCIN1	input	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌														🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌											
	CLCIN2	input									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌											🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌					
	CLCIN3	input									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌											🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌					
I2C1 ▼	SCL1	in/out									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌						🔌	🔌	🔌	🔌	🔌	🔌	🔌											
	SDA1	in/out									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌												
OSC	CLKOUT	output							🔌																																	
Pin Module ▼	GPIO	input	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌				
	GPIO	output	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌				
RESET	MCLR	input																																				🔌				
TMR4	T4IN	input									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌										🔌				
TMR6	T6IN	input									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌						
UART2 ▼	CTS2	input									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌										🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌						
	RTS2	output									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌						
	RX2	input									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌								🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌						
	TX2	output									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌								🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌					
	TXDE2	output									🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌								🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌	🔌				

Fig. 16 - Maschera di MCC relativa al "Pin Manager (Grid View)".

Pin Name	Module	Function	Custom Name	Start High	Analog	Output	WPU	OD	IOC
RA0	CLC2	CLCIN0		☐	☐	☐	☐	☐	none
RA0	CLC1	CLCIN0		☐	☐	☐	☐	☐	none
RA3	Pin Module	GPIO	AK9750_PDN	☐	☐	☒	☐	☐	none
RA4	Pin Module	GPIO	LED_D2	☐	☐	☒	☐	☐	none
RA5	Pin Module	GPIO	LED_D3	☐	☐	☒	☐	☐	none
RA6	Pin Module	GPIO	LED_D4	☐	☐	☒	☐	☐	none
RA7	Pin Module	GPIO	LED_D5	☐	☐	☒	☐	☐	none
RB1	UART2	TX2		☐	☐	☒	☐	☐	none
RB5	Pin Module	GPIO	AK9750_INT	☐	☐	☐	☐	☐	none
RB7	UART2	RX2		☐	☐	☐	☐	☐	none
RC3	I2C1	SCL1		☐	☐	☒	☐	☒	none
RC4	I2C1	SDA1		☐	☐	☒	☐	☒	none
RC5	CLC2	CLCIN1		☐	☐	☐	☐	☐	none
RC5	CLC1	CLCIN1		☐	☐	☐	☐	☐	none

Fig. 17 - Maschera di MCC relativa al "Pin Module".

In questo modo avremo creato due funzioni, una che ci restituirà in un `uint32_t` il numero di millisecondi trascorsi dall'ac-

Listato 1

```
/**
 * User defined variables
 */

volatile uint32_t Timer_millis = 0;

/**
 * End of User defined variables
 */

/**
 * User defined functions
 */

uint32_t millis(void) {
    uint32_t m;
    //
    PIE9bits.TMR6IE = 0; // Disable Timer Interrupt
    m = Timer_millis;
    PIE9bits.TMR6IE = 1; // Enable Timer Interrupt
    //
    return m;
}

void delay(uint32_t delayTime) {
    uint32_t tmpTime = millis();
    while (!(millis() - tmpTime >= delayTime)) {
    }
}

/**
 * End of User defined functions
 */
```

censione (funzione `millis()`) ed un'altra che ci permetterà di inserire nel codice dei ritardi con precisione del millisecondo (`delay()`); (chi lavora o ha lavorato in ambito Arduino, non avrà difficoltà a riconoscere le suddette funzioni che sono messe a disposizione dal framework "wiring" e sarà già abituato ad utilizzarle).

Occorrerà ora passare alla modifica/scrittura di tutte le parti del `main()` che permettono l'implementazione della nostra applicazione. Il codice è qui riportato, per ragioni di compattezza, al netto di una parte dei commenti che introduce MCC ma che, naturalmente, troverete nei file che scaricherete dall'area download della rivista. La prima parte della modifica riguarda l'inserimento delle righe del **Listato 3** all'inizio del programma, prima della funzione `main()` vera e propria. Le prime due `#define` servono solo per facilitare l'eventuale modifica delle tempistiche del super-loop ed indicano, la prima la cadenza di aggiornamento dei LED e la seconda la cadenza di lettura del sensore AK9750, entrambe espresse in millisecondi.

Segue una serie di `#include`, alcune inserite originalmente da MCC, altre aggiunte manualmente al codice generato. Sempre a seguire si trova la dichiarazione di una serie di variabili globali utilizzate dal programma.

Terminata la dichiarazione delle variabili si è inserita una funzione che ha il compito di preparare un buffer di caratteri, alternando scritte costanti con i valori letti dal sensore, buffer che alla fine, in funzione del valore di uscita della CLC2, viene inviato sulla seriale o tramite DMA (e quindi troviamo le necessarie assegnazioni ai registri e l'avvio del trasferimento) o

Listato 2

```
void TMR6_Callback(void)
{
    // Add your custom callback code here
    uint32_t m = Timer_millis;
    m += 1;
    Timer_millis = m;

    // this code executes every TMR6_INTERRUPT_TICKER_FACTOR periods of TMR6
    if(TMR6_InterruptHandler)
    {
        TMR6_InterruptHandler();
    }
}
```

Listato 3

```
#define LED_UPDT    50UL
#define AK_UPDT    750UL
#include <xc.h>
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include "mcc_generated_files/mcc.h"
#include "drvAK975x.h"

// Global variables

uint16_t irVal_1;
uint16_t irVal_2;
uint16_t irVal_3;
uint16_t irVal_4;
float temperature;
char s_Buffer[400];
char s_Riga[45];

void printData(void) {
    bool    fOutput;
    uint16_t sLen;
    //
    fOutput = CLC2_OutputStatusGet();
    sprintf(s_Buffer, "%cMicrochip Curiosity HPC with PIC18LF46K42", 0x01);
    sprintf(s_Riga, "%c%cIR1: %6u, IR2: %6u", 0x0D, 0x0D, irVal_1, irVal_2);
    strcat(s_Buffer, s_Riga);
    sprintf(s_Riga, "%c%cIR3: %6u, IR4: %6u", 0x0D, 0x0D, irVal_3, irVal_4);
    strcat(s_Buffer, s_Riga);
    if (fOutput) {
        sprintf(s_Riga, "%c%cTemp: %3.2f", 0x0D, 0x0D, temperature);
    } else {
        sprintf(s_Riga, "%c%cTemp: %3.2f", 0x0D, 0x0D, temperature);
    }
    strcat(s_Buffer, s_Riga);
    sLen = strlen(s_Buffer);
    //
    if (fOutput) {
        // Initialize DMA ...
        // Set the source address

        DMA1SSA = (uint24_t) s_Buffer;
        // Set the source size
        DMA1SSZ = sLen;
        // Set working mode
        DMA1CON0bits.SIRQEN = 1;
        DMA1SIRQ = 51;
        DMA1CON1bits.DSTP = 0;
        DMA1CON1bits.SSTP = 1;
        // Enable and Start DMA
        DMA1CON0bits.EN = 1;
        DMA1CON0bits.DGO = 1;
    } else {
        printf(s_Buffer);
    }
}
```


Listato 4

```

/*
                                     Main application
*/
void main(void) {
    uint8_t retValue = 0;
    uint8_t conta = 0;
    uint32_t lastMillisLED = 0;
    uint32_t lastMillisAK = 0;
    // Initialize the device
    SYSTEM_Initialize();
    // If using interrupts in PIC18 High/Low Priority Mode you need to enable the Global High and Low Interrupts
    // If using interrupts in PIC Mid-Range Compatibility Mode you need to enable the Global Interrupts
    // Use the following macros to:
    // Enable the Global Interrupts
    INTERRUPT_GlobalInterruptEnable();
    // Disable the Global Interrupts
    //INTERRUPT_GlobalInterruptDisable();
    // Init Curiosity HPC LEDs
    LED_D2_SetDigitalOutput();
    LED_D3_SetDigitalOutput();
    LED_D4_SetDigitalOutput();
    LED_D5_SetDigitalOutput();
    LED_D2_SetLow();
    LED_D3_SetLow();
    LED_D4_SetLow();
    LED_D5_SetLow();
    // Init AK9750 PDN and INT pins
    AK9750_PDN_SetDigitalOutput();
    AK9750_INT_SetDigitalInput();
    AK9750_PDN_SetHigh();
    delay_ms(100);
    // Start Timer6 (used for millis() function)
    TMR6_Start();
    retValue = AK975x_Init();
    if (retValue == FALSE) {
        printf("%cUnable to start AK9750, program halted.", 0x18);
        for (;;)
    }
    if (AK975x_available()) {
        irVal_1 = AK975x_readIR1();
        irVal_2 = AK975x_readIR2();
        irVal_3 = AK975x_readIR3();
        irVal_4 = AK975x_readIR4();
        temperature = AK975x_getTemperature();
        AK975x_refresh();
    }
    printf("%cCuriosity HPC initialized and started ...", 0x18);
    delay(3000);
    printf("%c ", 0x18);
    delay(500);
    while (1) {
        // Add your application code
        if (millis() - lastMillisLED > LED_UPDT) {
            if (conta & 0x01) LED_D2_SetHigh();
            else LED_D2_SetLow();
            if (conta & 0x02) LED_D3_SetHigh();
            else LED_D3_SetLow();
            if (conta & 0x04) LED_D4_SetHigh();
            else LED_D4_SetLow();
            if (conta & 0x08) LED_D5_SetHigh();
            else LED_D5_SetLow();
            conta++;
            if (conta > 15) conta = 0;
            lastMillisLED += LED_UPDT;
        }
        //
        if (millis() - lastMillisAK > AK_UPDT) {
            if (AK975x_available()) {
                irVal_1 = AK975x_readIR1();
                irVal_2 = AK975x_readIR2();
                irVal_3 = AK975x_readIR3();
                irVal_4 = AK975x_readIR4();
                temperature = AK975x_getTemperature();
                AK975x_refresh();
                printData();
            }
            lastMillisAK += AK_UPDT;
        }
    }
}
/**
End of File
*/

```

tramite la funzione `printf()`.

Facciamo notare che in tale funzione, volutamente, si sono utilizzati nella formattazione parecchi spazi (anche dove non occorrevano o dove sarebbe stato possibile utilizzare la capacità di indirizzamento riga/colonna del terminale), questo per avere un numero significativo di bytes da trasmettere e quindi, nel caso di utilizzo della `printf()`, impegnare la CPU per un tempo apprezzabile ad occhio umano, rallentando l'aggiornamento dei LED.

La seconda parte della modifica riguarda invece l'inserimento della porzione di codice riportata nel **Listato 4** all'interno della funzione `main()` vera e propria. All'inizio del `main()` vengono prima di tutto dichiarate una serie di variabili locali che verranno utilizzate all'interno dello stesso. Segue l'inizializzazione di tutte le CIP da parte di MCC e l'abilitazione degli interrupt (che vengono da noi utilizzati per il Timer6).

A seguire troviamo l'inizializzazione degli I/O relativi ai LED e la chiamata alle necessarie funzioni per l'inizializzazione del sensore AK9750. Viene poi avviato il Timer6, verificato il funzionamento del sensore AK9750 (in caso di malfunzionamento di detto sensore, il programma viene arrestato in un loop senza fine) ed acquisiti i primi valori dalle sue quattro celle IR. A conclusione di questa fase di inizializzazione troviamo l'invio di un messaggio sul terminale seriale e, dopo una breve attesa, la sua pulizia per prepararlo alla ricezione dei dati letti.

Terminata la fase di inizializzazione troviamo il super-loop (`while(1) { ... }`) nel cui interno vengono verificate due temporizzazioni; la prima verifica se sono passati LED_UPDT millisecondi ed in caso affermativo incrementa un contatore e lo visualizza sui LED, la seconda, effettuata ogni AK_UPDT millisecondi, effettua la lettura del sensore AK9750 e richiama la funzione di visualizzazione dei dati sul terminale seriale. Compilato e caricato il programma sulla "Curiosity HPC", la collegheremo al TinyTerminal (**Fig. 18**) mediante il connettore



Fig. 18 - L'hardware con cui testare il programma.

mikroBUS 2 in questo modo:

- RX del terminale su TX della Curiosity;
- +5V del terminale su +5V della Curiosity;
- GND del terminale su GND della Curiosity.

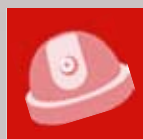
Alimentata la scheda e avviato il firmware, vedrete apparire il messaggio di inizializzazione sul terminale e dopo una breve attesa, i LED inizieranno a contare in binario con periodicità 50 ms ed il terminale seriale a visualizzare i valori letti dalle quattro celle IR del sensore AK9750. Nel caso sia selezionata la `printf()`, ovvero la scrittura sulla UART via software a carico della CPU, noterete che l'avanzamento regolare del conteggio binario, sarà interrotto ogni 750 ms causa l'impegno della suddetta CPU, mentre, selezionando il DMA (... con la pressione del tasto S2 si alterna tra CPU e DMA), si vedrà chiaramente che la trasmissione dei dati seriali non rallenterà il conteggio binario, essendo fatta parallelamente al lavoro della CPU dalla CIP DMA.

CONCLUSIONI

Con questa settima ed ultima puntata si conclude il nostro corso sulle CIP (e sull'utilizzo di MCC per la loro programmazione) che possiamo trovare in alcune delle MCU a 8 e 16 bit di Microchip. Ovviamente il corso non è esaustivo e su altre MCU potremmo trovare altre CIP di cui qui non abbiamo discusso, ma, capito ormai il meccanismo di funzionamento, non sarà difficile per il lettore, attraverso lo studio del datasheet e l'aiuto di MCC, utilizzarle nei propri programmi. Una nota finale: MCC è un potente strumento che aiuta e velocizza nella configurazione di varie MCU e delle loro periferiche (che siano CIP o meno), ma è pur sempre uno strumento scritto da "uomini" che, come tali, nonostante la massima attenzione posta, possono comunque commettere qualche errore.

Se qualcosa non funziona, controllate, con l'ausilio del datasheet, che i vari sorgenti di MCC siano corretti e facciano realmente quanto avete richiesto; se riscontrate qualche anomalia (un registro che, per quanto da voi selezionato dovrebbe assumere un determinato valore, mentre viene inizializzato con un altro) non esitate a segnalare la cosa al supporto Microchip. MCC è in continuo sviluppo e quindi le vostre segnalazioni saranno gradite e le correzioni prontamente applicate. Non dimenticate poi che, proprio perché esso è in continuo sviluppo, è possibile che una MCU da voi selezionata non sia ancora supportata; Microchip dà ovviamente priorità più alta all'implementazione delle MCU più recenti e di quelle che mano a mano vengono annunciate e, con priorità più bassa, implementa le MCU più datate.

Ricordiamo infine ai tutti i lettori che hanno avuto modo di apprezzare le potenzialità degli MCU con CIP, che è disponibile un corso specifico presso "Futura Accademy" (<https://futura.academy/corsi/pic8-cip-core-independent-peripherals-e-mcc-microchip-code-configurator/>). □



Sfoggia on-line i cataloghi 2019

**FUTURA
ELETTRONICA®**



**PRESTO
DISPONIBILI
ANCHE:**





SUPER-Lion, batterie più performanti per il mondo connesso

Con il supporto di un programma Marie Curie, il dott. Knut Bjarne Gandrud ha sviluppato un elettrolita super sottile allo stato solido per le batterie ricaricabili.

Le batterie agli ioni di litio (LIB, Lithium Ion Batteries) attualmente funzionano con elettroliti liquidi. Il nuovo tipo di fabbricazione degli elettroliti allo stato solido di SUPER-Lion potrebbe tuttavia aprire la strada al loro uso in batterie sottili per i dispositivi che vengono sviluppati per il mercato emergente dell'Internet delle cose. Lo spessore di 40 nanometri dell'elettrolita super sottile lo rende ideale per i microdispositivi.

La ricerca del dott. Gandrud è stata supervisionata dal professor Philippe Vereecken, direttore scientifico presso l'hub dedicato a nanoelettronica e tecnologie digitali IMEC, e professore alla KU-Leuven in Belgio, ed è stata assistita da due studenti di master.

Le LIB che sfruttano elettroliti liquidi sono usate nella maggior parte dei dispositivi quali telefoni cellulari e laptop, ma anche nelle automobili elettriche.

La maggior parte delle batterie con litio allo stato solido tendono a essere usate in applicazioni di minori dimensioni che non necessitano di molta energia o dove la sicurezza è fondamentale, come ad esempio i dispositivi medici impiantabili.

Il progetto ha sviluppato un elettrolita composto su nanoscala (NCE, Nano Composite Electrolyte) che può essere integrato facilmente in batterie più piccole, consentendo di incrementare facilmente l'energia della cella della batteria. Il metodo di fabbricazione è inoltre facile da portare su scala industriale per la produzione di massa al giusto prezzo.

<https://cordis.europa.eu>

SpaceX, pieno successo per il primo lancio commerciale di Falcon Heavy

È stato effettuato dalla storica piattaforma 39A del Kennedy Space Center di Cape Canaveral, in Florida, il secondo lancio del razzo Falcon Heavy di SpaceX. Dopo il primo lancio di test, questa volta si è trattato di una missione commerciale vera e propria, col vettore SpaceX che ha portato in orbita il satellite per telecomunicazioni Arabsat-6A, appartenente all'omonima emittente araba che offre centinaia di canali televisivi in tutto il Medio Oriente e il Nord Africa. Il lancio è stato coronato da successo in ogni suo aspetto, col satellite che è stato messo regolarmente in orbita e con i due booster e il primo stadio vettore che sono rientrati in atmosfera e regolarmente atterrati, pronti – dopo una revisione – ad essere riutilizzati in una nuova missione.

Il Falcon Heavy consiste in un Falcon 9 con l'aggiunta di ulteriori due primi stadi come booster, una configurazione simile all'EELV Delta IV, all'Atlas V HLV e al Russo Angara A5V. Il razzo è stato progettato per raggiungere e superare i criteri minimi per lanciare esseri umani. I margini di sicurezza sono superiori del 40% rispetto alle sollecitazioni normali che si incontrano durante i lanci, rispetto al 25% degli altri lanciatori. Il Falcon Heavy è stato progettato per portare astronauti verso l'orbita terrestre bassa e ripristinare la possibilità di raggiungere la Luna e, in futuro, Marte. Non solo: la tecnologia riutilizzabile consentirà di ridurre di 2-3 volte il costo della messa in orbita di carichi pesanti.

www.spacex.com



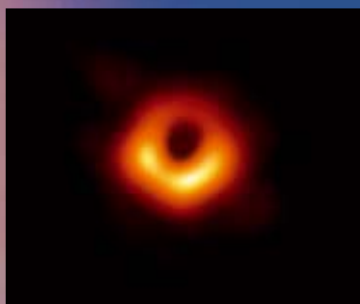
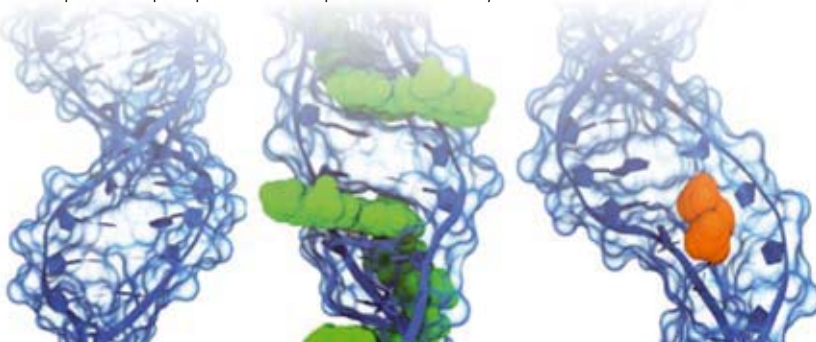
Nanorisonatori meccanici di DNA

Il DNA, il codice genetico che racchiude le basi della vita, è una sequenza di informazioni necessarie per il funzionamento di tutte le molecole che compongono un organismo. D'altra parte, come dimostrato da diversi studi recenti, il DNA non è solo un codice passivo, ma interagisce attivamente anche con le molecole esterne, le quali possono modificare profondamente la sua struttura chimica. E proprio di questo aspetto tratta la ricerca pubblicata sulla rivista *Nature Communications* "Nanomechanical DNA resonators for sensing and structural analysis of DNA-ligand complexes" dal team di cui fanno parte Stefano Stassi e Carlo Ricciardi, ricercatore e docente del Dipartimento di Scienza Applicata e Tecnologia del Politecnico di Torino. La modalità di interazione con diverse tipologie di biomolecole è stata individuata grazie all'utilizzo di risonatori nanomeccanici: sono state create delle nanofibre di DNA – ovvero più

fasci di DNA "attorcigliati" che creano una "corda" più resistente rispetto alla singola molecola – che, vengono messe in oscillazione fino a "suonare" la propria frequenza di risonanza. E in base alla molecola con cui interagiscono, le nanofibre di DNA risuonano a diverse frequenze, come fossero le "note" che scaturiscono dalle "corde" di uno strumento. Comprendere e regolare il meccanismo di interazione del DNA con il maggior numero di molecole possibile può portare ad importanti

sviluppi, in un futuro prossimo, nell'ambito medico. Si potrebbe infatti intervenire su più fronti: dal dosaggio più corretto di farmaci invasivi come i chemioterapici – da somministrare in quantità precise e circoscritte alle cellule colpite da malattia – o ancora comprendere come possano essere indotte mutazioni nel DNA a causa dell'interazione con ambienti inquinati da metalli pesanti.

www.polito.it



Scattata la prima foto di un buco nero

L'Event Horizon Telescope, collaborazione internazionale che vede la partecipazione di centri di ricerca in tutto il mondo, ha realizzato la foto del secolo.

Due ricercatrici dell'Inaf, Elisabetta Liuzzo e Kazi Rygl, sono tra i protagonisti della rivoluzionaria osservazione del gigantesco buco nero nel cuore della galassia Messier 87, come parte del progetto BlackHoleCam. Un altro italiano, Ciriaco Goddi, è segretario del consiglio scientifico del consorzio Eht e responsabile scientifico del progetto BlackHoleCam.

L'Event Horizon Telescope (Eht) è un gruppo di otto radiotelescopi da terra che opera su scala planetaria, nato grazie a una collaborazione internazionale e progettato con lo scopo di catturare le immagini di un buco nero. I ricercatori dell'Eht hanno annunciato il successo del progetto, svelando la prima prova visiva diretta mai ottenuta di un buco nero supermassiccio e della sua ombra.

Questo incredibile risultato viene presentato in una serie di sei articoli pubblicati in

un numero speciale di *The Astrophysical Journal Letters*.

L'immagine rivela il buco nero al centro di Messier 87, un'enorme galassia situata nel vicino ammasso della Vergine. Questo buco nero dista da noi 55 milioni di anni luce e ha una massa pari a 6,5 miliardi e mezzo di volte quella del Sole.

L'Eht collega gli otto radiotelescopi dislocati in diverse parti del pianeta dando vita a un telescopio virtuale di dimensioni pari a quelle della Terra, uno strumento con una sensibilità e una risoluzione senza precedenti. L'Eht è il risultato di anni di collaborazione internazionale e offre agli scienziati un nuovo modo di studiare gli oggetti più estremi dell'universo previsti dalla teoria della relatività generale di Einstein, proprio nell'anno del centenario dell'esperimento storico che per primo ha confermato questa teoria.

www.inaf.it

Auto a guida autonoma, al via la sperimentazione anche in Italia

A breve, anche sulle strade italiane, potremo incrociare veicoli senza conducente. La sperimentazione dei veicoli senza conducente sulle strade pubbliche è infatti stata approvata dal Ministero delle Infrastrutture e dei Trasporti il 7 maggio. La Direzione generale Motorizzazione ha accolto il parere positivo dell'Osservatorio tecnico di supporto per le Smart Road rilasciato lo scorso 22 marzo e deciso di permettere alla società VisLab Srl la sperimentazione delle auto a guida autonoma tra Torino e Parma. Il ministro Danilo Toninelli su Facebook ha commentato: "Questo Ministero guarda al futuro e alle nuove tecnologie per la sicurezza e la serenità di chi viaggia".

Il Ministero dei Trasporti in una nota diffusa il 7 maggio ha spiegato di aver concluso le necessarie verifiche preliminari di idoneità tecnica alla circolazione dell'auto a guida autonoma con esito positivo e ha deciso di dare il via libera alla sperimentazione alla VisLab Srl, una società di Parma; VisLab, spiegano dal ministero, ad oggi è l'unica ad aver presentato la domanda di autorizzazione. La sperimentazione si svolgerà a Torino e Parma, riguarderà l'ambito urbano e l'ultimo miglio 'tipo D, E, F' in precisi tratti stradali delle città.

www.mit.gov.it



Aerei commerciali, arrivano i giapponesi

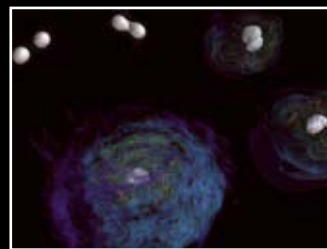
Potrebbe essere l'aereo giusto al momento giusto: i giapponesi ci stanno provando col primo aereo di linea costruito in Giappone dagli anni '60, che ha iniziato i voli di certificazione lo scorso mese a Moses Lake, a Washington. Mentre il Mitsubishi Regional Jet sta effettuando i voli di test, Embraer e Bombardier, i due rivali nel campo dei voli regionali, stanno per essere acquistati da Boeing e Airbus. Un mercato, quelli dei jet regionali fino a 160 posti che vale oltre 135 miliardi di dollari nei prossimi 20 anni. Con pochi posti a sedere e fusoliere più piccole, i jet regionali sono una classe diversa di velivoli da aerei più grandi

a corpo stretto come il Boeing 737 o l'A320 dell'Airbus. L'MRJ ha un'autonomia di circa 2.000 miglia ed è in grado di trasportare 76 o 88 passeggeri. Da lungo tempo fornitore di componenti per aeromobili Boeing, Mitsubishi inizialmente ha programmato voli di prova nel 2012 ma ha superato questa scadenza a causa di difficoltà di produzione. Ora l'azienda, che produce navi, centrali nucleari e componenti aerospaziali, prevede di avere l'aereo pronto per i clienti l'anno prossimo, un calendario che metterà alla prova la società, ha dichiarato il presidente di Mitsubishi Aircraft Corp. Hisakazu Mizutani.

www.mhi.com



Onde Gravitazionali, nuovi dati raccolti da LIGO-Virgo



Il primo mese della nuova campagna di osservazione O3 dei tre interferometri LIGO-Virgo, iniziata il 1° aprile, e prevista per durare un anno, sta già premiando con un grande numero di segnali candidati di onde gravitazionali l'impegno scientifico e lo sforzo tecnologico delle collaborazioni LIGO-Virgo, che ora stanno lavorando all'interpretazione dei dati raccolti. I miglioramenti ottenuti nella sensibilità dei rivelatori e il fatto che i tre strumenti, i due interferometri LIGO negli Stati Uniti (nello stato di Washington e in Louisiana), e Virgo in Italia, siano stati operativi simultaneamente dal primo giorno, stanno offrendo un'opportunità senza precedenti per lo studio delle onde gravitazionali. Inoltre, nella nuova campagna O3 LIGO-Virgo diramano allerta pubblici che ora sono subito disponibili, per facilitare le osservazioni di follow-up da parte di altri telescopi, e migliorare così il potenziale delle osservazioni multimessaggere. In questo primo mese di attività sono stati lanciati cinque allerta pubblici (che possono essere consultati nel database degli eventi candidati di onde gravitazionali).

"Virgo funziona con la massima stabilità: sta osservando il cielo per il 90% del tempo, contribuendo così in modo fondamentale al puntamento delle sorgenti", spiega Giovanni Prodi, ricercatore dell'INFN e professore all'Università di Trento, coordinatore dell'analisi dei dati di Virgo.

www.infn.it

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



**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

Prezzi IVA inclusa.

**FUTURA
ELETTRONICA®**

www.futurashop.it

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

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

Bosch e le celle a combustibile per autocarri e automobili

Bosch entra nel mercato delle celle a combustibile e apre la strada per l'adozione di questa tecnologia negli autocarri e nelle automobili. Uno dei componenti fondamentali del sistema è la pila, l'anima della cella a combustibile che converte l'idrogeno in energia elettrica. Per migliorarle ulteriormente e realizzare queste pile, Bosch ha siglato un accordo con Powercell Sweden AB, il produttore svedese di pile a combustibile. I due partner lavoreranno insieme per rendere la cella a combustibile PEM (a membrana elettrolitica polimerica) pronta per la produzione. Bosch produrrà questa tecnologia di prodotti per il mercato automotive a livello globale. La pila a combustibile completerà l'offerta Bosch di componenti per le celle a combustibile e verrà lanciata entro il 2022.

"Nel campo delle celle a combustibile, Bosch ha già una buona presenza che verrà ulteriormente rafforzata da questa cooperazione." ha dichiarato Stefan Hartung, membro del Board of Management di Bosch.

www.bosch.it



Verso un mondo di edifici ad alta efficienza energetica

Gli edifici sono responsabili approssimativamente del 40% del consumo energetico e del 30% delle emissioni di biossido di carbonio nell'UE. Ma che cosa si può fare per ridurre queste statistiche preoccupanti? Un progetto finanziato dall'UE ha la soluzione. L'ecosistema energetico informatico di ENTROPY sfrutterà, a questo scopo, l'opportunità legata alle possibilità di una piattaforma basata sull'Internet degli oggetti, dell'analisi dei grandi dati, di raccomandazioni personalizzate.

La soluzione si attiva quando un responsabile avvia una campagna di efficienza energetica attraverso la piattaforma ENTROPY. Da lì, si specificano le aree dell'edificio e si riceve in cambio una serie di sensori installati per area. Successivamente, i sistemi di

Terna, Eni, CDP e Fincantieri per l'energia da moto ondoso su scala industriale

Gli amministratori di Terna, ENI, CDP e Fincantieri hanno firmato all'EUR di Roma, un accordo per lo sviluppo e la realizzazione su scala industriale di impianti di produzione di energia dalle onde del mare.

L'accordo ha lo scopo di unire le competenze di ciascuna società al fine di trasformare il progetto pilota Inertial Sea Wave Energy Converter (ISWEC), l'innovativo sistema di produzione di energia dal moto ondoso installato da Eni nell'offshore di Ravenna e attualmente in produzione, in un progetto realizzabile su scala industriale e quindi di immediata applicazione e utilizzo.

Secondo i termini dell'accordo, Eni metterà a disposizione del gruppo di lavoro congiunto i risultati dell'impianto pilota ISWEC, sviluppato in sinergia con il Politecnico di Torino e lo spin-off Wave for Energy, e fornirà il proprio know-how tecnologico, industriale e commerciale, oltre a rendere disponibili le opportunità logistiche e tecnologiche dei propri impianti offshore.

CDP promuoverà il progetto con le pub-

bliche amministrazioni e le istituzioni coinvolte e, inoltre, metterà a disposizione le proprie competenze economico-finanziarie, anche al fine di valutare le più

adeguate forme di supporto finanziario dell'iniziativa.

Fincantieri offrirà le proprie competenze industriali e tecniche tipiche delle realiz-





monitoraggio dei dati vengono attivati e memorizzati nel deposito di ENTROPY in modo standardizzato.

Non appena raccolti questi dati, viene fornita una serie di servizi. Il responsabile della campagna produce analisi energetiche e comportamentali. Successivamente, sulla base dell'analisi dei risultati e del profilo di ciascun utente, vengono inviati servizi e messaggi personalizzati a ciascun utente. Gli utenti finali usano quindi i servizi di ENTROPY attraverso un'applicazione personalizzata sul proprio smartphone.

<https://entropy-project.eu/>



zazioni navali per l'ottimizzazione delle fasi di progettazione esecutiva, costruzione e installazione delle unità di produzione.

Terna contribuirà a sviluppare gli studi relativi alle migliori modalità di connessione e integrazione del sistema di produzione di energia con la rete elettrica, ivi inclusa l'integrazione con i sistemi ibridi composti da generazione convenzionale, impianti di produzione fotovoltaici e sistemi di stoccaggio.

www.terna.it

Fusione nucleare: i 100 secondi che possono fare la storia

Tanto è durata la reazione nucleare nel nuovo reattore a fusione cinese EAST (Experimental Advanced Superconducting Tokamak), che si trova nella provincia orientale dell'Anhui. Il plasma dello stesso reattore aveva raggiunto alla fine del 2018 la temperatura record di 100 milioni di gradi, cinque volte quella del sole.

Tali temperature da capogiro sono cruciali per ottenere reazioni di fusione nucleare sostenibili, che promettono una fonte di energia inesauribile.

Un passo storico se si pensa che in passato, nei reattori sperimentali che erano riusciti ad innescare la fusione, il plasma era rimasto attivo per frazioni di millisecondo. Un risultato che sottolinea la determinazione di Pechino ad essere al centro della tecnologia per l'energia pulita, con una road map che prevede un impianto completamente funzionante entro il 2050.

EAST fa parte del progetto internazionale Thermonuclear Experimental Reactor (ITER), che cerca di dimostrare la fattibilità della fusione.

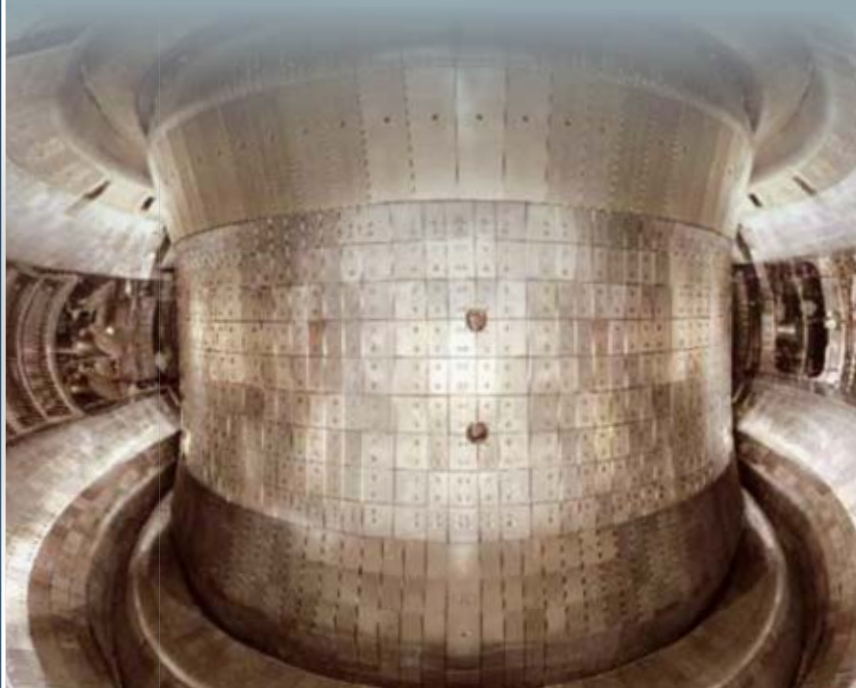
"Speriamo di espandere la cooperazione internazionale attraverso questo impianto (EAST) e di dare contributi cinesi al futuro uso dell'umanità della fusione nucleare", ha detto Song Yuntao, un alto funzionario coinvolto nel progetto.

Miliardi di dollari sono stati promessi per l'ambizioso progetto. Finanziato e gestito dall'Unione Europea, dall'India, dal Giappone, dalla Cina, dalla Russia, dalla Corea del Sud e dagli Stati Uniti, il fulcro del progetto multimiliardario sarà un gigantesco dispositivo di fusione cilindrica, chiamato tokamak.

La fusione è considerata il Santo Graal, il nostro sole. Unisce i nuclei atomici per creare enormi quantità di energia - l'opposto del processo di fissione utilizzato nelle armi atomiche e nelle centrali nucleari, che le divide in frammenti.

A differenza della fissione, la fusione non emette gas ad effetto serra e comporta meno rischi di incidenti o il furto di materiale radioattivo.

<https://m.phys.org>



AIR RACE E, per lo sviluppo della propulsione elettrica nei cieli

Non ci sono solo le auto elettriche e ibride: la stessa tecnologia può essere implementata anche in campo aeronautico. E per far sviluppare il "volo elettrico", il gigante aerospaziale Airbus ha annunciato una partnership con Air Race E, la prima gara al mondo di aeroplani elettrici che prenderà il via nel 2020.

Airbus è partner e fondatore di Air Race E, una competizione che mira a spingere lo sviluppo e l'adozione di motori elettrici più puliti, veloci e tecnologicamente avanzati su aeromobili commerciali e privati. Come ha affermato Grazia Vittadini, Chief technology officer di Airbus: *"Vogliamo motivare i produttori a mettere in mostra le loro tecnologie che riguardano i sistemi di propulsione elettrica con i vari componenti. Questa partnership ci consente di dimostrare il nostro impegno nel rimanere all'avanguardia nella propulsione elettrica e nello sviluppo di nuove tecnologie".*

Come la Formula E sta alla Formula 1, la Air Race E sta alla Formula 1 Air Racing. Otto aerei a propulsione elettrica si sfideranno l'uno contro l'altro su un circuito di 5 km, volando a soli 10 metri dal suolo ad una velocità superiore a quella

di qualsiasi veicolo terrestre. Per sviluppare il velivolo Airbus sta collaborando con altri partner, tra cui l'Università di Nottingham che sta realizzando il prototipo di un aereo a propulsione elettrica che aiuterà a creare il modello e a stilare il regolamento del futuro campionato.

Il CEO di Air Race E, Jeff Zaltman, ha dichiarato: *"Non potremmo essere più felici o più ottimisti per il successo con*

Airbus come nostro partner fondatore ufficiale. Questa partnership è una pietra miliare significativa per l'evoluzione della potenza elettrica nell'aviazione. Insieme, stiamo lavorando per creare un sistema in cui l'innovazione nella propulsione elettrica possa essere sviluppata, alimentata e accelerata più rapidamente".

<https://airracee.com/>



L'impianto fotovoltaico olandese che insegue il sole

Floating Solar, una società olandese, ha iniziato la costruzione del più grande impianto fotovoltaico galleggiante che, in più, è in grado, grazie ad un sistema di boe, di inseguire il movimento del sole, tecnologia che garantisce un rendimento superiore del 30%. Il sito sorgerà nella riserva naturale di Andijk, nel Nord dell'Olanda, e sarà composto da 15 isolotti, per un totale di 73.500 moduli fotovoltaici. Il progetto prenderà il via appena conclusa

la stagione delle migrazioni degli uccelli acquatici con la costruzione dei primi 3 isolotti, ognuno dei quali avrà un diametro di 140 metri. Il termine della prima fase dei lavori è previsto per novembre 2019. Una volta completato, l'arcipelago fotovoltaico galleggiante produrrà energia elettrica per almeno 10 mila abitazioni: un livello superiore del 30% con i pannelli che si muovono da est ad ovest grazie ad un semplice meccanismo realizzato con tre

boe. Lo stesso meccanismo sarà utile per proteggere gli isolotti da condizioni meteo estreme: un apposito sistema di gestione (weather risk management – WRM) farà ruotare gli isolotti nel caso rilevasse un'eccessiva esposizione al vento permettendo così alle onde di passare attraverso i pannelli senza arrecare danni. Il WRM è stato testato con venti fino a 60 miglia orarie.

<https://floatingsolar.nl>



Tutto sotto controllo ovunque e in ogni momento!



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

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

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

€ 56,⁰⁰
cod. CP729



€ 54,⁰⁰
cod. CP728

Istruzioni
in italiano!

TELECAMERA IP WiFi MOTORIZZATA da 2MPX ONVIF

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





MOUSER
ELECTRONICS.



 **SamacSys**
ELECTRONIC COMPONENT LIBRARY SOLUTIONS

Scarica GRATUITAMENTE

- Simboli schematici
- Footprint per PCB
- Modelli 3D

Clicca e rilascia nel tuo strumento CAD

Prodotti d'avanguardia per progetti innovativi™

mouser.it/ecad