

Elettronica In

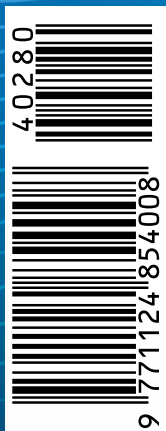
www.elettronica.in.it

€ 7,50 - Bimestrale, anno XXX - n. 280 - AGOSTO / SETTEMBRE 2024

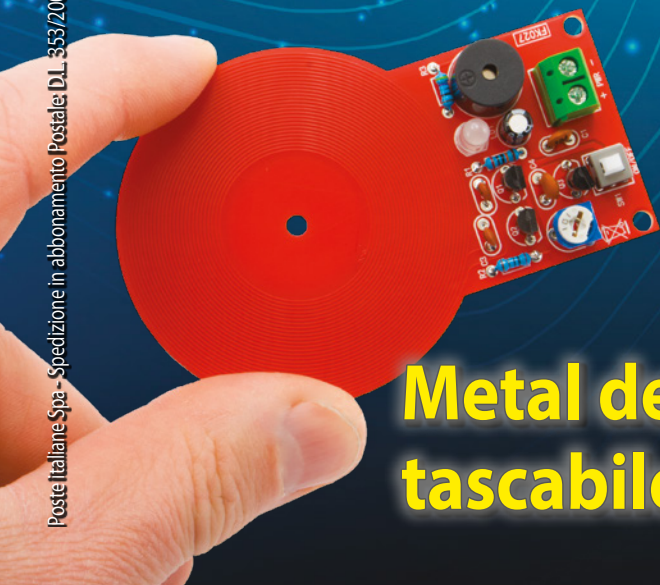
Telecomando universale con gesture



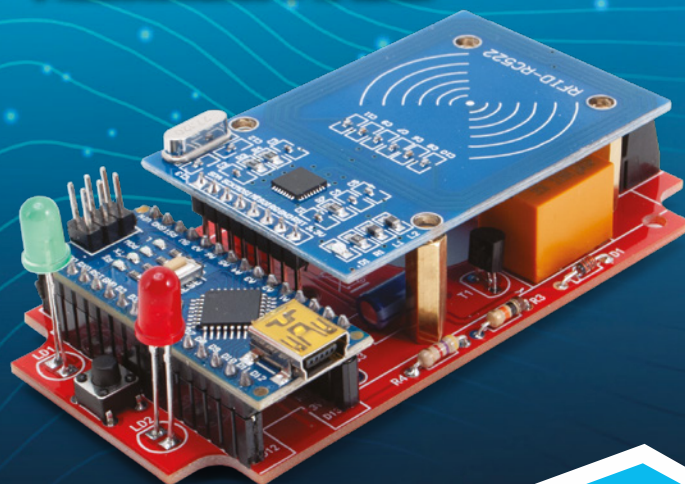
- ▶ Amplificatore HI-FI 2X15W
- ▶ Smart House con Arduino
- ▶ Bersaglio intelligente
- ▶ Breakout con ATtiny85
- ▶ Programmazione microcontrollori ARM
- ▶ Comunicazione Wireless flessibile
- ▶ Gestore di alimentazione
- ▶ Radiocomando a 12 canali
- ▶ Arduino e gli automi
- ▶ Monitor di rete elettrica



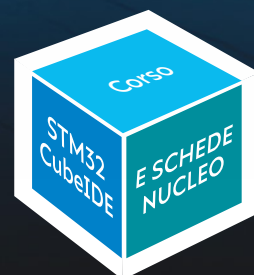
Poste Italiane SpA - Spedizione in abbonamento Postale (DL 353/2003 (conv. in L. 27/02/2004 n.46) art. 1 - comma 1 - DCB Milano. Prima immissione: 02/08/2024.



Metal detector tascabile



Serratura RFID



ROME

Maker Faire

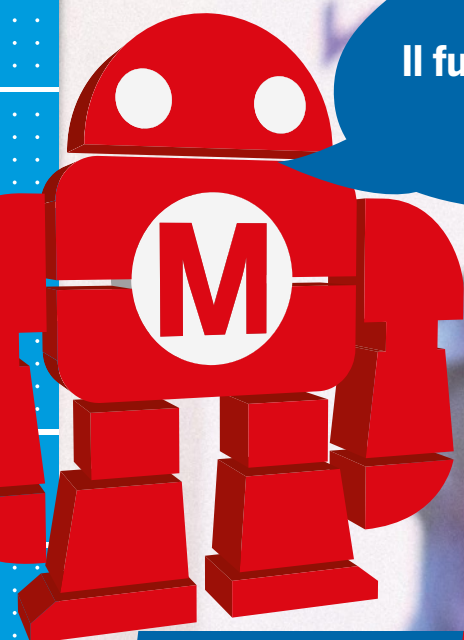
THE EUROPEAN EDITION

An event powered by



Camera di Commercio
Roma

Il futuro si incontra
a Roma!



25-27
OTTOBRE
2024

SCOPRI, INNOVA, CREA!
GAZOMETRO OSTIENSE ROMA

MAKERFAIREROME.EU - #MFR2024

Promosso
e organizzato da



Camera di Commercio
Roma

Gold Partner



DigiKey

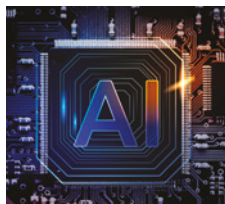


Silver Partner



Bronze Partner





L'Intelligenza Artificiale rivoluziona (anche) l'elettronica

L'intelligenza artificiale (AI) sta trasformando rapidamente il campo dell'elettronica, sia consumer che industriale, migliorando le prestazioni e l'efficienza dei dispositivi. Ad esempio, i computer stanno diventando più capaci, ottimizzando le prestazioni e riducendo il consumo energetico grazie all'AI. Nell'industria, i cobot, robot collaborativi, stanno migliorando significativamente i processi di assemblaggio e

controllo qualità, offrendo maggiore precisione e sicurezza grazie all'AI. Anche nel mondo dell'Internet delle cose (IoT), l'intelligenza artificiale viene utilizzata per gestire la grande mole di dati prodotta dai dispositivi connessi. Come possiamo noi sviluppatori integrare e sperimentare la potenza dell'AI nelle nostre

Esistono diversi strumenti messi a disposizione da chi ha investito nell'intelligenza artificiale

applicazioni? Fortunatamente esistono diversi strumenti messi a disposizione da chi ha investito nell'intelligenza artificiale. Google, ad esempio, ha presentato già nel 2019 MediaPipe, una suite di librerie e strumenti per applicare rapidamente tecniche di AI e ML. In questo numero di Elettronica In, troverete un progetto dettagliato che sfrutta MediaPipe con un Raspberry Pi per riconoscere i gesti della mano, permettendo di comandare dispositivi come la televisione semplicemente muovendo la mano davanti alla telecamera. Il progetto descritto è tra i primi di una lunga serie che esploreremo nei prossimi numeri di Elettronica In. Vi invito quindi a non perdere le future edizioni, dove continueremo a presentare soluzioni innovative e applicazioni dell'AI che stanno spingendo i confini della tecnologia. L'era dell'AI nell'elettronica è appena iniziata e le possibilità che offre sono infinite.

ON LINE

[elettronica.in](mailto:elettronica.in@elettronica.in)

SI RINGRAZIA

Futura Elettronica
Microchip
Mouser
Win Source



In copertina:
Il telecomando
dotato di Intelligenza
Artificiale, gestito con
Raspberry Pi e la suite
MediaPipe di Google.

Boris Landoni
 boris.landoni@elettronica.in

Anno XXX n. 280
AGOSTO / SETTEMBRE 2024

EDITORIALE

Direttore Responsabile

Boris Landoni
 boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta
 redazione@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)
 monica.premoli@elettronica.in

Ufficio Abbonamenti

Tel. 0331-752668
 abbonati@elettronica.in

Direzione, Redazione, Pubblicità

FUTURA GROUP srl - Divisione Editoriale
via Adige 11 - 21013 Gallarate (VA)
 Tel. 0331-752668

Abbonamenti

Annuo 6 numeri Euro 39,00

Annuo 6 numeri digitale Euro 29,00

Le richieste di abbonamento vanno inviate a:

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

Tel. 0331-752668

oppure tramite il sito

<https://futuranet.it/abbonamenti>

Distribuzione per l'Italia

SODIP srl
via Bettola 18 - 20092 Cinisello Balsamo (MI)
 Tel. 02-660301 Fax 02-66030320

Stampa

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



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

Prezzo di copertina Euro 7,50.

Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili
sul sito della rivista al prezzo di Euro 7,50.

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 (per maggiori informazioni
scrivere a redazione@elettronica.in). Manoscritti, disegni e
foto non richiesti non verranno in alcun caso restituiti. L'utilizzo
dei progetti e dei programmi pubblicati non comporta alcuna
responsabilità da parte della Società Editrice.

© 2024 Futura Group srl

Futura
Group
Edizioni

Contenuti

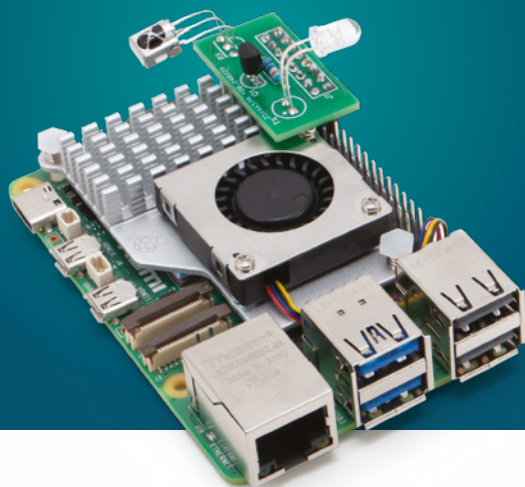
01

RUBRICHE

- 01 Editoriale
- 05 T&T Tips & Tricks
- 156 Scienza & Tecnologia

Progetto di Copertina

TELECOMANDO UNIVERSALE AI MEDIAPIPE PER TV

**73**

AUDIO

Finale Hi-Fi 2x15W

**11**

GADGET

Smart Target

**19**

PROGRAMMAZIONE

Programmazione microcontrollori ARM

**29****11**

ARTICOLI

AUDIO

11 Finale Hi-Fi 2X15W

Amplificatore di potenza in grado di sonorizzare senza problemi un locale o l'abitacolo di un'automobile pilotando due casse acustiche da 8 ohm.

GADGET

19 Smart Target

Realizziamo bersagli avanzati, interconnessi in una rete, ideali per competizioni con pistole giocattolo. L'intero sistema è facilmente gestibile tramite smartphone, grazie all'integrazione del modulo ESP32.

PROGRAMMAZIONE

29 Programmazione microcontrollori ARM

Esploriamo metodi per programmare microcontrollori ARM, utilizzando strumenti economici e adattamenti pratici, rendendoli accessibili agli appassionati di elettronica hobbistica.

AUTOMAZIONE

41 Latch di alimentazione

Permette di alimentare circuiti elettronici con la semplice pressione di un pulsante, intervenendo direttamente sull'alimentazione principale e consumando nulla quando è a riposo.

HOME PROJECT

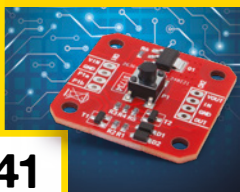
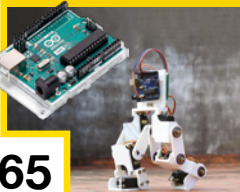
47 Mini metal detector

Cercametri di piccole dimensioni ideale sia per ritrovare monete e oggetti metallici nella terra e nella sabbia, sia per rilevare corpi metallici nei vestiti e nelle borse o tubi di ferro nei muri fino a 6 cm.

SICUREZZA

55 Serratura RFID

Operante sulla frequenza standard di 13,56 MHz, viene governato da una piccola Arduino Nano per aprire e chiudere un relé in base al riconoscimento dei tag avvicinati al modulo RFID.

AUTOMAZIONE**Latch di alimentazione****41****HOME PROJECT****Mini metal detector****47****SICUREZZA****Serratura RFID****55****PROGRAMMAZIONE****Arduino e gli automi a stati finiti****65****SICUREZZA****Monitor di rete****85****PROGRAMMAZIONE****65 Arduino e gli automi a stati finiti**

Scopriamo le macchine a stati finiti, utili in automazione e hobbistica per un design chiaro e semplice, fondamentale in elettronica e automazione industriale.

INTELLIGENZA ARTIFICIALE**73 Telecomando universale AI MediaPipe PER TV**

Realizziamo un dispositivo basato su Raspberry Pi capace di riconoscere gesti predefiniti della mano ed associarli a comandi ad infrarosso.

SICUREZZA**85 Monitor allarme di assenza rete elettrica**

Scoraggiamo i malintenzionati che ci staccano la corrente, con un dispositivo che attiva l'allarme se manca la tensione di rete per più di qualche secondo, con conteggio degli eventi occorsi.

DIDATTICA**95 Smart House con Arduino**

Realizziamo una Smart House automatizzata con sensori per luci, temperatura, umidità e altri parametri ambientali, gestiti da una scheda compatibile con Arduino UNO.

DIDATTICA**107 Breakout con ATtiny85**

Scopri il potenziale del microcontrollore ATtiny85 in applicazioni pratiche come la gestione di encoder rotativi e PID, sfruttando l'efficiente collegamento I2C.

I PROGETTI DI eLektor**117 Comunicazione wireless flessibile**

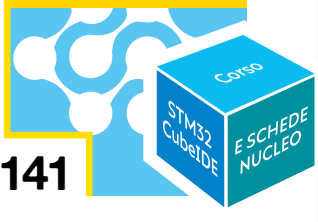
Quando si connette un microcontrollore a una rete Wi-Fi utilizzando l'ESP8266, potrebbe essere utile adottare un approccio più flessibile rispetto all'uso di credenziali WLAN hard-coded e fisse. Offriamo una soluzione con un insieme di reti AP preferite da scegliere in modo interattivo e sfruttiamo la funzionalità Wi-Fi Protected Setup (WPS) supportata da molti AP e router.

MODELLISMO**125 Radiocomando 12 canali**

Dotato di 12 uscite a relè azionabili da telecomandi a 4 e a 12 tasti, questo ricevitore può funzionare sia in modo tradizionale a singolo relè, che in modo "fermodellismo" a combinazione di sequenze.

DIDATTICA**141 STM32CubeIDE e schede Nucleo: la potenza dell'embedded a portata di mano**

Analizziamo la configurazione Pulse Width Modulation del Timer, implementando un progetto per variare la luminosità di un LED e introduciamo la periferica UART per la gestione della trasmissione e ricezione dati. Quarta puntata.

DIDATTICA**Smart House con Arduino****95****DIDATTICA****Breakout con ATtiny85****107****eLektor****Comunicazione MCU flessibile****117****MODELLISMO****Radiocomando 12 canali****125****DIDATTICA****STM32CubeIDE e schede Nucleo****141**



Migliora i Sistemi Spazio e Difesa con le MCU SAMD21RT

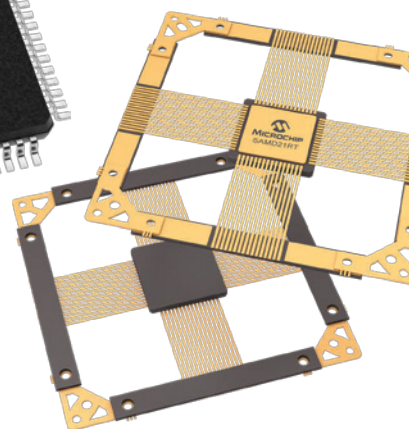
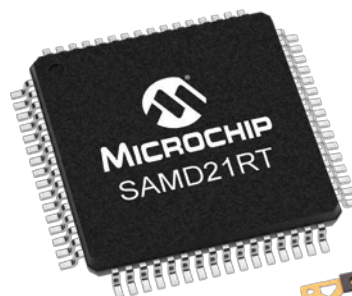
Piccolo Ingombro, Radiation Tolerant, Arm® Cortex® -M0+

Cerchi una soluzione per valorizzare i tuoi progetti per lo Spazio e Difesa? Non cercare oltre, i microcontroller SAMD21RT offrono una gamma di vantaggi, su misura, per soddisfare le tue esigenze.

Catturando tutta la potenza di un processor core a 32 bit, consentono di sviluppare anche per gli spazi più ristretti. Con un ingombro compatto di soli 10 x 10 mm, infatti, le MCU SAMD21RT offrono prestazioni semplicemente eccezionali. Il nostro obiettivo è progettare prodotti che risultino economici, di piccole dimensioni ed efficienti dal punto di vista energetico per rivoluzionare i tuoi dispositivi.

I nostri microcontroller SAMD21RT ti offrono risorse e strumenti a tuo vantaggio:

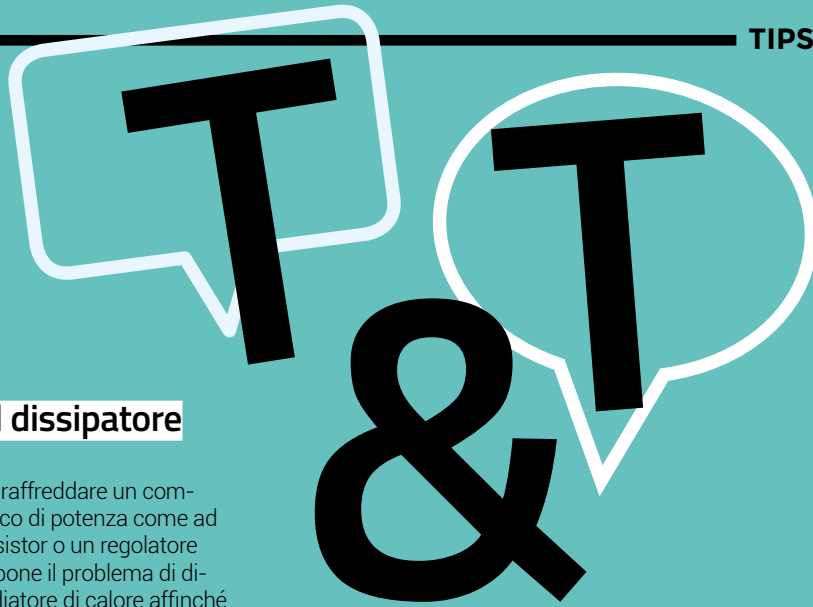
- Elevato livello di tolleranza alle radiazioni con capacità TID fino a 50 kRad e immunità SEL fino a 78 MeV. cm²/mg
- MPLAB X Integrated Development Environment – programma software espandibile e altamente configurabile
- MPLAB® PICKit™ 5 in-circuit debugger/programmer
- SAM D21 Curiosity Nano Evaluation Kit



microchip.com/SAMD21RT



Il nome e logo Microchip e il logo Microchip sono marchi industriali registrati di Microchip Technology Incorporated negli U.S.A. e in altri Stati. Tutti gli altri marchi appartengono ai rispettivi titolari.
© 2024 Microchip Technology Inc. Tutti i diritti riservati.
MEC2573A-ITA-06-24



Calcolare il dissipatore

Quando bisogna raffreddare un componente elettronico di potenza come ad esempio un transistor o un regolatore di tensione, ci si pone il problema di dimensionare il radiatore di calore affinché impedisca al componente stesso di superare la massima temperatura operativa ammessa; lo scopo del radiatore è, infatti, mantenere a una certa temperatura il semiconduttore, idealmente almeno una decina di gradi centigradi sotto la massima temperatura di giunzione specificata nel datasheet dal costruttore come parametro T_{jmax} .

Ma come calcolare il dissipatore e quali caratteristiche e parametri bisogna prendere in considerazione? Ebbene, si parte dal problema che è quello di smaltire il calore sviluppato dal chip (altrimenti detto "die") di semiconduttore; il calore è un'energia che deve andare dalla zona più calda ad una meno calda, perché diversamente non può essere smaltito, vale a dire allontanato dall'area che lo produce: la zona calda è il semiconduttore e quella fredda è, nel caso dei dissipatori di calore "ad aria" ossia le comuni "alette di raffreddamento" come usiamo chiamarli, l'ambiente esterno. Affinché la dissipazione avvenga, dunque, l'ambiente deve essere più freddo del componente e questo è ciò che si verifica normalmente; inoltre la facilità con la quale tale calore viaggia dalla fonte (il semiconduttore) all'ambiente esterno dipende da una grandezza fisica nota come "resistenza termica" ed espressa con la lettera greca Teta (θ). Tale resistenza termica può essere considerata l'opposto di quella che oggi va tanto di moda in ambito di edilizia a contenimento energetico e che è nota come "trasmissanza": mentre quest'ultima si esprime in $W/^{\circ}C$, la resistenza termica si misura in $^{\circ}C/W$. Un dissipatore da $10^{\circ}C/W$ di resistenza termica presenta, in analogia con la Legge di Ohm, una caduta di temperatura (meglio parlare di salto termico...) di $10^{\circ}C$ ogni watt di potenza termica che lo attraversa.

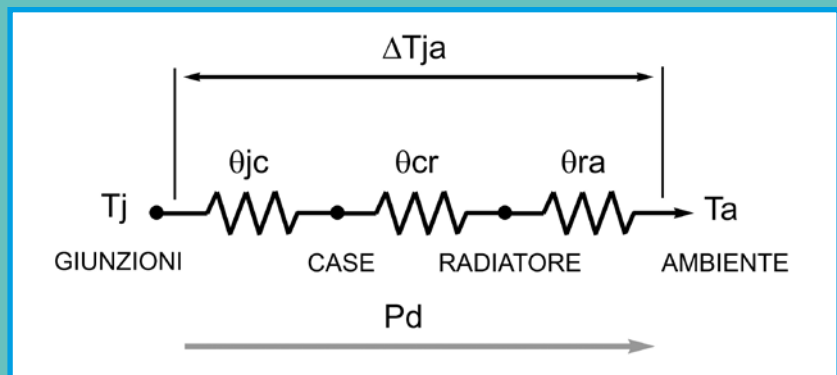
Il produttore di ogni componente a semiconduttore definisce, oltre alla T_{jmax}

(che può essere $150^{\circ}C$, $175^{\circ}C$ o $200^{\circ}C$) la resistenza termica complessiva tra semiconduttore e ambiente, relativa al funzionamento del componente senza dissipatore (heatsink) e quella tra semiconduttore e corpo (tipicamente la zona metallica retrostante, come ad esempio nei package TO3P, SOT-93, TO-126 e, TO220) del componente: la prima viene chiamata θ_{ja} (thermal resistance junction to ambient) mentre la seconda è la θ_{jc} (thermal resistance junction to case) e viene definita per una determinata temperatura del contenitore. Bene, la resistenza termica complessiva che il calore incontra per uscire dal chip e disperdersi nell'ambiente può essere vista

come una resistenza elettrica e nel calcolo del dissipatore possiamo applicare la Legge di Ohm, considerando la potenza da dissipare (P_d) come la corrente e la differenza di temperatura come la caduta di tensione (DV) ottenendo quindi:

$$\theta_{ja} = DV / P_d$$

La resistenza termica complessiva è la somma di più resistenze termiche, ossia, come schematizzato nell'immagine in questa pagina, quella tra semiconduttore (giunzioni) e case, quella tra case e radiatore e quella tra radiatore e ambiente. Quest'ultima è la resistenza termica propria del dissipatore (θ_{ra}) che bisogna



↑ Rappresentazione della resistenza termica che il calore incontra nel percorso di smaltimento.

calcolare in base alla potenza da dissipare e al salto termico desiderato, imposta una massima T_j ammissibile.

Invece la θ_{cr} è la resistenza termica insita nell'accoppiamento fra parte metallica del componente a semiconduttore e radiatore di calore e dipende sia dal tipo di case, sia da cosa è frapposto tra il componente e il dissipatore; a titolo d'esempio, per un contenitore TO-220 la θ_{cr} vale $0,5 \text{ } ^\circ\text{C/W}$ per l'accoppiamento mediante pasta termica e $1 \text{ } ^\circ\text{C/W}$ interponendo un foglietto isolatore di mica o teflon grigio.

Proviamo a chiarire i concetti prendendo ad esempio un transistor TIP35, che tollera una T_{jmax} di $150 \text{ } ^\circ\text{C}$ e che può dissipare, se gli si mantiene il contenitore a $25 \text{ } ^\circ\text{C}$, 125 W ; ha quindi una resistenza termica di:

$$\theta = \Delta T / P_d = (150 - 25) \text{ } ^\circ\text{C} / 125 \text{ W} = 125 \text{ } ^\circ\text{C} / 125 \text{ W} = 1 \text{ } ^\circ\text{C/W}$$

Supponiamo di farlo lavorare a una temperatura ambiente di $35 \text{ } ^\circ\text{C}$ e di arrivare al massimo a una T_j di $150 \text{ } ^\circ\text{C}$, cosicché la differenza di temperatura vale $115 \text{ } ^\circ\text{C}$. Determiniamo quindi la resistenza termica totale come:

$$\theta = \Delta T / P_d = (T_{jMAX} - T_a) / P_d$$

Imponendo che il TIP35 debba dissipare $35,7 \text{ W}$ si ha:

$$\theta = (T_{jMAX} - T_a) / P_d = (150 - 35) \text{ } ^\circ\text{C} / 35,7 \text{ W} = 3,22 \text{ } ^\circ\text{C/W}$$

Dato che la θ ottenuta è la somma di tutte le resistenze termiche, si può ottenere la sola θ_{ra} , ossia la resistenza termica del dissipatore di calore, dalla relazione:

$$\theta_{ra} = \theta - \theta_{jc} - \theta_{cr}$$

La resistenza termica tra contenitore (θ_{cr}) e radiatore del package TO3-P in cui è incapsulato il nostro TIP35 vale $0,3 \text{ } ^\circ\text{C/W}$ se il transistor è appoggiato direttamente al dissipatore o $0,7 \text{ } ^\circ\text{C/W}$ se viene interposto un foglietto isolatore. Immaginate, dunque, di montare il TIP35 senza isolamento e con l'interposizione della sola pasta al silicone; nota la θ_{jc} e stabilito che la θ_{cr} vale $0,3 \text{ } ^\circ\text{C/W}$, si ricava la θ_{ra} :

$$\theta_{ra} = \theta - \theta_{jc} - \theta_{cr} = (3,22 - 1 - 0,3) \text{ } ^\circ\text{C/W} = 1,92 \text{ } ^\circ\text{C/W}$$

Questo è il valore di resistenza termica massimo ammesso per il dissipatore da utilizzare.

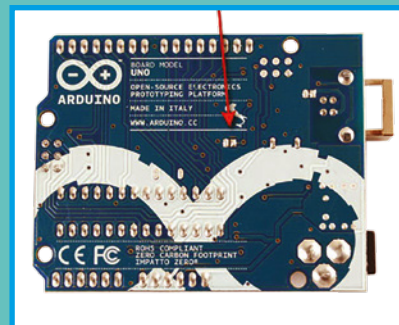
Hackerare l'USB di Arduino

Con l'avvento della UNO, il team di Arduino ha sostituito il tradizionale converter TTL/USB della FTDI con un microcontrollore Atmega 8u2, che rispetto a un tradizionale converter permette di fare tante cose; con il programma predefinito, ossia il firmware con cui la Uno viene programmata, l'ATmega 8u2 emula il chip della FTDI, creando un'interfaccia USB/RS232, però più gestibile anche in ambiente Macintosh perché simula un dispositivo seriale CDC generico.

In ambiente Windows è comunque richiesto un file .inf, ma non è necessario installare alcun driver, il che mette al riparo dai problemi di compatibilità con le varie versioni del sistema operativo.

La cosa interessante dell'ATmega 8u2 è che essendo un microcontrollore può essere riprogrammato per farlo funzionare come vari Device USB (l'Host rimane sempre il computer cui Arduino si collega): ad esempio può funzionare come una tastiera o un mouse, quindi una periferica HID, oppure un'unità a disco, un'interfaccia MIDI ed altro ancora.

Per riprogrammare l'ATmega 8u2 bisogna saldare un connettore ISP a 6 pin (Arduino UNO R3 ha già a bordo tale connettore) e collegarlo a un programmatore AVR standard; è anche possibile usare il bootloader (DFU) nell'ATmega8u2. Sulle board Arduino UNO di prima generazione, lo si abilita saldando una resistenza da 10 kohm tra due piazzole collocate dal lato saldature della scheda, che si trovano dove mostrato dalla foto in questa pagina.



↑ Le due piazzole da unire con il resistore da 10 kohm per abilitare il bootloader in Arduino UNO.

Nelle versioni successive, ossia la R2 e la R3 della UNO, viene montato tipicamente un microcontrollore ATmega 16u2 al posto dell'8u2 e questo non richiede il resistore; pertanto nelle più recenti Arduino UNO non vi sono più le piazzole mostrate nella foto. Il codice per l'8u2 è basato su LUFA, il fantastico USB-AVR di Dean Cameron libreria che ha ottimi esempi e documentazione. È anche completamente open source.

Paneling e depaneling

Quando bisogna realizzare molti esemplari di circuiti stampati tutti uguali, è conveniente accorparli in un unico layout multiplo per poi inciderli e lavorarli su una singola piastra di vetronite, separandoli alla fine o dopo il montaggio dei componenti; questa tecnica prende il nome di depaneling e si svolge in due fasi principali: nella prima (panelizzazione o



↑ Separazione dei singoli circuiti stampati con taglio mediante lama a disco.

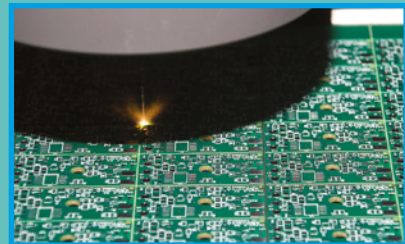
paneling che dir si voglia) si crea il layout multiplo con i relativi file gerber per la produzione e nella seconda si provvede alla separazione dei singoli PCB una volta incisi, forati e trattati per la serigrafia, il solder-resist, la placcatura ecc. Naturalmente questa tecnica implica l'adozione di una serie di accorgimenti, in primis nella fase di progettazione del layout da cui creare i file gerber: non basta semplicemente accostare le singole tracce lato rame e le disposizioni componenti (silk) ma tra i singoli elementi bisogna prevedere un certo spazio che dipende da come verrà poi separato dagli altri adiacenti ciascun PCB elementare. Tale spazio dipende dalla tecnica utilizzata per il depaneling: la maniera più semplice per separare più moduli realizzati su una grande piastra ramata consiste nel creare solchi lungo i bordi dei singoli PCB, ad esempio con una fresa, quindi piegare la fila più esterna bloccando bene la parte rimanente con ancora su le file di PCB non ancora separate; fatto ciò, le file così ottenute si inviano a un procedimento analogo per separare da esse i moduli individualmente. Analogamente è possibile utilizzare una lama spinta con forza sulla superficie della piastra lungo il bordo, ma non operando a taglio, bensì a compressione; anche in questo caso, che nella pratica è una punzonatura, aver realizzato un solco lungo i bordi facilita la procedura. In luogo del solco è possibile ricavare, in fase di foratura del PCB, una fila di fori lungo i bordi del singolo elemento, eventualmente intervallata da cave fresate sempre durante la foratura; questa tecnica è più consigliabile quando si opera con piastre basate su materiali un po' fragili come le bacheliti e simili. Linee preforate consentono anche il depaneling per taglio mediante lame a disc operanti sia sopra che sotto il panel, impiegando macchine sia manuali che automatiche. Un'altra tecnica di depaneling è il taglio mediante seghe circolari, ma in quest'ultimo caso, oltre ad avere polveri e sfridi da asportare durante la lavorazione o al suo termine, è possibile effettuare solo tagli dritti e si pongono limitazioni, per il depaneling di PCB-A, all'altezza dei componenti sul lato dove opera la sega. È anche possibile separare i PCB mediante taglio a laser, che oggi è una delle tecniche migliori. A prescindere dalla tecnica utilizzata, nel progettare le pellicole e i telai di serigrafia occorre tenere conto dello scarto, ossia quanto materiale va considerato che si perda tra le frazioni; per esempio, se si pratica un solco largo 1 mm tra i bordi di file adiacenti di PCB e le file che prendono posto su una piastra sono 6, si perderan-

no in tutto 5 mm. Sulle lastre e sui layout di foratura i singoli PCB andranno quindi distanziati di 1 mm. Se si ricorre a cave fresate e file di fori accostati, bisognerà distanziare i singoli circuiti stampati di almeno il diametro dei fori o la larghezza delle cave. Il routing è la tecnologia industriale più utilizzata per separare i punti di attacco dei PCB al termine della produzione e si applica sia ai semplici circuiti stampati che ai PCB-A. Nel router depaneling, un utensile taglia i punti di separazione dei singoli circuiti stampati fresandoli, senza causare alcuno stress meccanico e questo vale ancor più per la produzione dei PCB-A, che vengono separati quando sono già popolati dai componenti. Grazie a questo processo, infatti, i punti di saldatura dei componenti vicini al bordo tecnico non subiscono stress che possano danneggiarli o esporli a malfunzionamenti futuri. Per elevare il livello di qualità della lavorazione, si adotta un sistema di aspirazione integrato attorno all'utensile e controllato dalla macchina, che evita di lasciare sfridi sui singoli circuiti stampati. La fresatura avviene ad opera di macchine a controllo numerico eventualmente assistite da sistemi di visione artificiale, applicando sopra la piastra ramata una maschera già fresata in alluminio che guida l'utensile nel percorso di fresatura delle cave, evitando che sconfini in aree dei PCB che devono rimanere integre. La piastra ramata viene fissata ben centrata a un piatto in alluminio la cui superficie riporta cave analoghe a quelle da realizzare, ma più larghe, per lasciare muovere la fresa senza che faccia danni, che sforzi o, peggio ancora, che sollevi trucioli parassiti. Sopra la piastra viene applicata la maschera, anch'essa in alluminio, che guida la fresa nel proprio lavoro; la maschera è centrata rispetto al piatto di alluminio. Un sistema molto interessante per la separazione dei PCB dalla piastra originaria è quello che si basa sul taglio a laser (tipicamente a CO₂ e quindi molto potente) i cui pregi sono l'alta qualità, la flessibilità e i costi contenuti, oltre che la maggiore semplicità meccanica rispetto alle macchine per il depaneling meccanico. Infatti a differenza dei processi di depaneling meccanico come fresatura o punzonatura, il taglio a laser non lascia polvere sulle schede pertanto non richiede la relativa pulizia (soddisfando gli standard relativi alla pulizia tecnica, come VDA 19/ISO 16232) né implica l'asportazione di polveri e trucioli; inoltre non richiede maschere, dime e via di seguito.

Un ulteriore vantaggio dell'utilizzo del laser è che consente l'ablazione strato per strato, quindi la stessa macchina permette non solo la separazione dei PCB, ma anche la rimozione di singoli layer o spessori di vetronite.

La tecnica a laser è versatile perché non è necessario che i bordi dei PCB siano dritti né sono imposti raggi minimi di curvatura, linguette o limitazioni tipici del depaneling meccanico, ad esempio per fresatura. Il laser offre una precisione di taglio superiore e quindi canali di taglio molto stretti, consentendo di limitare fortemente le perdite di materiale del panel, fino al 30% rispetto al depaneling meccanico. Inoltre l'elevata potenza delle sorgenti laser odierne consente di raggiungere velocità di taglio competitive rispetto ai sistemi meccanici.

Un altro esempio di azienda produttrice di macchine per il depaneling a laser è la statunitense CMLASER, che realizza sistemi assistiti anch'essi da apparati di visione TTV fuori asse, con strumento di taglio a laser ultravioletto o verde, ad anidride carbonica raffreddato ad acqua; possono effettuare il depaneling sia su piastre di materiale comune come l'FR4 (la comune vetronite) e il teflon, sia su PCB flessibili e sottili. Anche questa azienda sviluppa macchine sia stand-alone che in-line.



↑ Separazione a laser dei PCB.

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**



ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in.it

6 NUMERI
A SOLI

€29

ANZICHÉ € 45,00

ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

nella pagina "**Abbonati o rinnova**"
del nostro sito www.elettronicaain.it

→ **e-mail**

scrivendo a
abbonati@elettronicaain.it

→ **telefono**

telefonando a
+39-0331-752668

NUOVO FORMATO 160 PAGINE CON PIÙ CONTENUTI

ABBONAMENTO ANNUALE PER RIVISTA **CARTACEA**

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

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

BUONO SCONTO DA 5€ valido 12 mesi
da utilizzare su **Futurashop.it** nelle categorie "Libri" o "Riviste"

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

6
NUMERI
A SOLI

€39

ANZICHÉ € 45,00

Selezione completa di componenti elettronici

Principale distributore di componenti elettronici



Componenti regolari e parti obsoleti

WIN SOURCE è orgogliosa di essere classificata al 16° posto tra i distributori globali di componenti elettronici. Guidati dalle tecnologie IoT, AI e di automazione, abbiamo stabilito un vasto sistema di Catena di fornitura globale e un sistema di dati intelligente, affrontando efficacemente l'asimmetria delle informazioni nella Catena di fornitura e fornendo ai clienti un'interfaccia di acquisto efficiente e facile da usare.



FINALE HI-FI

2X15W



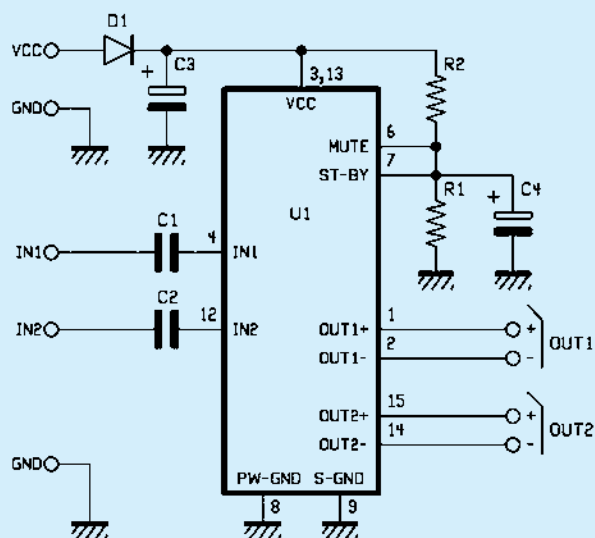
di **DAVIDE SCULLINO**

Amplificatore di potenza in grado di offrire un'esperienza sonora di alta qualità in un locale o nell'abitacolo di un'automobile, pilotando due casse acustiche da 8 ohm.

N

el panorama della riproduzione musicale, i circuiti integrati "la fanno ormai da padrone" e non solo per quel che riguarda le funzioni più complesse legate all'audio digitale (CODEC) ma ormai anche nella sezione di amplificazione e quindi quella di potenza.

Qui sono da tempo apparsi amplificatori integrati dalle prestazioni eccellenti, specie in fatto di potenza d'uscita in rapporto al consumo energetico, non a caso la loro efficienza li pone come componenti di elezione per gli apparati portatili, alimentati a batteria, dove il consumo è determinante. In commercio troviamo finali integrati in classe D, ma anche in classe H o classe T, che a volte lavorano anche senza bisogno di dissipatore di calore, almeno quando le potenze in gioco rimangono al disotto della decina di watt.



Questo genere di amplificatori, però, non è propriamente lineare, ovvero lo è se consideriamo che il segnale di uscita segue linearmente quello di ingresso, ma di fatto il segnale audio viene convertito in impulsi e poi riconvertito; questo, almeno, è quello che succede nella circuitazione in classe D, dove a ricostruire l'involuppo del segnale d'ingresso è un filtro LC (del second'ordine) posto tra l'uscita e l'altoparlante.

Chi non vuole rinunciare ad un amplificatore più tradizionale, beneficiando però della formula integrata, può trovare interessante il progetto descritto in queste pagine, che è un finale stereo integrato in classe AB, rivolto a tutti perché gli stadi occorrenti sono incorporati nel circuito integrato che lo realizza: il TDA7297 della ST.

Questo amplificatore stereo è molto efficiente poiché utilizza una tecnica speciale, chiamata configurazione a ponte, per il suo stadio di uscita. Questa configurazione, completamente integrata all'interno, combina due amplificatori per canale con un sistema che corregge la fase del segnale e regola automaticamente l'intensità del suono (guadagno). Grazie a ciò, l'amplificatore ha bisogno di pochissimi componenti aggiuntivi per funzionare. Questa caratteristica lo rende ideale per essere incluso in apparecchi audio sia domestici che portatili.

SCHEMA ELETTRICO

Andiamo quindi a vedere di cosa si tratta, analizzando lo schema elettrico di questo piccolo ma valido amplificatore stereo, dal quale si evince come tutti i componenti siano accorpati nell'integrato U1: il TDA7297, appunto.

Rimane esterna solamente la rete RC di temporizzazione richiesta dalle funzioni di Mute (silenzia-mento dell'uscita) e Standby (permette di spegnere tutti gli stadi interni di amplificazione pur mantenendo il circuito alimentato) di cui spiegheremo tra breve, che vanno ad arricchire la dotazione e la versatilità dell'amplificatore.

I segnali degli ingressi sinistro e destro si applicano ai contatti IN1 e IN2 rispettivamente e attraverso i condensatori di disaccoppiamento C1 e C2, necessari per bloccare la componente continua degli stadi preamplificatori impedendo che raggiunga la fonte BF stereo, arrivano ai piedini 4 e 12; a questi fanno capo gli ingressi (di tipo sbilanciato) degli stadi amplificatori di potenza ad elevato guadagno, che per ciascun canale sono virtualmente due operazionali collegati in modo da funzionare a ponte.

Con riferimento alla **Fig. 1**, dove vedete la struttura interna dell'integrato, ogni canale dispone di due operazionali con ingresso del segnale su quello in alto, configurato come non-invertente e retroazionato mediante una resistenza tra uscita e input invertente, più una resistenza di riferimento che fa partitore e che diviene, per lo stadio sottostante, la resistenza di ingresso. Lo stadio inferiore funziona in configurazione invertente.

La configurazione delle reti di retroazione è stata studiata per fare in modo che, conformemente ai requisiti di un amplificatore a ponte, i due operazionali presentino il medesimo guadagno in tensione e la stessa resistenza di uscita; nello specifico, il segnale di retroazione dell'operazionale superiore fa da ingresso a quello inferiore, grazie al resistore collegato tra i due input invertenti.

L'opportuno dimensionamento delle tre resistenze consente di ottenere dai due amplificatori praticamente lo stesso guadagno, fermo restando che siccome l'operazionale inferiore è in configurazione invertente la sua uscita presenterà un segnale di ampiezza uguale a quello uscente dall'operazionale superiore, ma di polarità e andamento opposti. Quindi a parità di segnale d'ingresso su IN1, il piedino OUT1+ presenterà un segnale di pari ampiezza rispetto a quello uscente da OUT1- ma di fase opposta; i due segnali sono riferiti a massa. Analogamente, per gli operazionali del canale destro (2) l'uscita OUT2+ presenterà un segnale di fase opposta rispetto all'uscita OUT2-, anche in questo caso con riferimento a massa.

Pertanto collegando gli altoparlanti dei singoli canali non verso massa ma tra le due uscite degli operazionali connessi a ponte, si ottiene una tensione di valore efficace praticamente doppia

rispetto a quella ottenibile collegando l'altoparlante tra un'uscita e massa.

Questo è il vantaggio della configurazione a ponte: poter sviluppare, a parità di ampiezza della tensione di alimentazione e di impedenza dell'altoparlante, una potenza teoricamente doppia rispetto a quella ottenibile con la configurazione single-ended, ossia collegando il carico tra l'uscita dell'amplificatore e la massa.

Un altro vantaggio della connessione a ponte è che permette di trascurare la componente continua di polarizzazione presente all'uscita dello stadio finale, perché siccome si presume che i due operazionali di potenza abbiano, a riposo, l'uscita a un potenziale pari a metà di quello di alimentazione (V_{cc}) l'altoparlante a riposo e in continua non riceve tensione, se non quel piccolo valore di offset (nel TDA7297 ammonta al massimo a 120 mV) che in un circuito a polarizzazione fissa non può essere evitato.

Per superare il problema della polarizzazione a riposo, in un classico amplificatore ad uscita singola (il single-ended...) bisogna ricorrere a un conden-

CARATTERISTICHE TECNICHE

◇ Tensione di alimentazione: 17V_{cc} ◇ Corrente assorbita: 2,5A ◇ Potenza massima su 8 ohm: 15+15W ◇ Risposta in frequenza: 20÷60 kHz ◇ Distorsione armonica totale: 0,1%	◇ Guadagno in tensione: 40 volte ◇ Segnale d'ingresso: 2x275 mV_{eff}. ◇ Impedenza d'ingresso: 30 kohm ◇ Separazione tra i canali: 60 dB ◇ Classe di funzionamento: AB
--	--

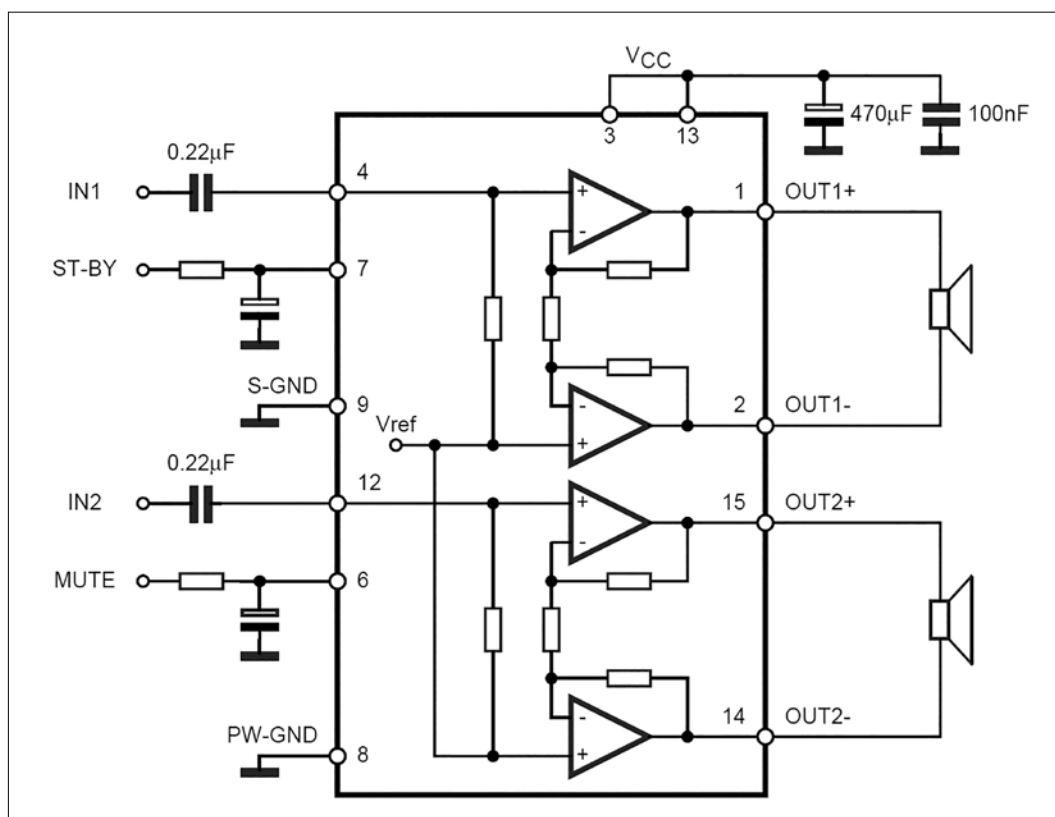
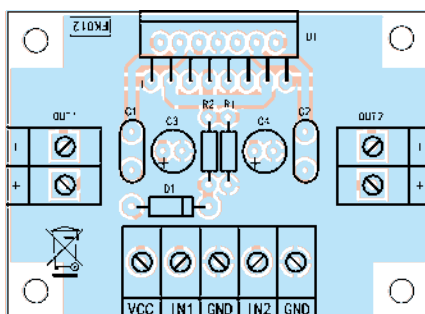
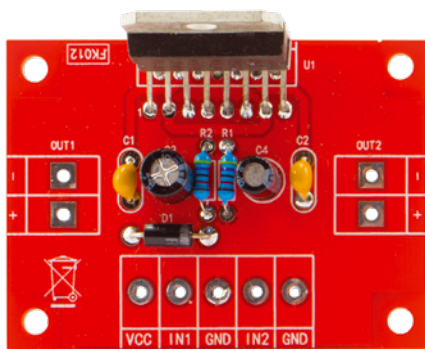


Fig. 1
Schema a blocchi e tipica applicazione del TDA7297.



Elenco Componenti:

R1, R2	47 kohm
C1, C2	220 µF multistrato
C3	100 µF 25 VL elettrolitico
C4	10 µF 25 VL elettrolitico
D1	1N4007
U1	TDA7297

Varie:

- Circuito Stampato FK012 (56x41 mm)

satore di accoppiamento posto in serie all'altoparlante oppure far funzionare lo stadio di potenza con un'alimentazione duale (ovvero simmetrica rispetto a massa); come vedete dallo schema elettrico, nel nostro caso adottiamo invece una sola alimentazione e ciò rende ideale il nostro circuito anche al funzionamento in auto o a batteria.

Siccome le reti di polarizzazione e di retroazione sono incorporate nel TDA7297, il guadagno in tensione è fisso e non si può variare dall'esterno; per l'esattezza il guadagno tipico è di 32 dB, ossia 40 volte in tensione e ciò significa che per ottenere la massima potenza (ossia i 15W per canale) su

carico da 8 ohm bisogna fornire a ciascun ingresso un segnale audio dell'ampiezza di 275 mVeff.

Come vedete, l'amplificatore non richiede le classiche reti di stabilizzazione RC ai capi delle uscite (le cosiddette Celle di Boucherot) perché la stabilità e la prevenzione dalle autoscillazioni vengono implementate internamente.

Torniamo adesso allo schema elettrico, dove vediamo che il TDA7297 è configurato nella maniera suggerita dal costruttore, con gli ingressi disaccoppiati mediante due condensatori e le uscite accoppiate in continua ciascuna coppia ad un altoparlante.

Vedete anche una rete RC che si collega ai piedini 6 e 7 dell'integrato, i quali fanno capo agli ingressi di comando delle funzioni, rispettivamente, di Mute e Standby: la prima permette di silenziare l'amplificatore di potenza (attenuando di 30 dB i segnali dei due canali) e può essere utilizzata ad esempio per abbassare completamente il volume senza togliere il segnale dagli ingressi, per esempio quando occorre rispondere al telefono o serve sentire altro che verrebbe coperto dalla musica riprodotta. Può anche essere utilizzata come è stato fatto nel nostro circuito, ossia per evitare fastidiosi "Pop-Noise" (ossia il tipico botto negli altoparlanti) e Click-Noise durante i transitori di accensione/spegnimento.

Allo scopo è necessario garantire la corretta sequenza di comando dei piedini di standby e Mute; la funzione standby permette invece di spegnere gli stadi di potenza pur lasciando alimentato l'amplificatore (che assorbe in tutto appena 100 microampere) e in questo caso permette di ottenere lo spegnimento e l'accensione mediante delle logiche. È abbastanza semplice ottenere questa funzione utilizzando un microprocessore, che può gestire la sequenza di comando dei piedini di Standby e di Mute.

Notate che le due funzioni sono attive quando i rispettivi piedini si trovano a zero logico, pertanto abbiamo collegato insieme i piedini 6 e 7 ad un partitore di tensione e ad un condensatore elettrolitico terminante a massa; in tal modo, quando si applica tensione al circuito lo stadio di potenza è inizialmente spento e privo di segnale d'ingresso e poi man mano che il C4 si carica viene liberato lo Standby e i circuiti si accendono.

Poco dopo, quando il condensatore continua a caricarsi puntando a una tensione pari a metà di quella che alimenta il TDA7297, esauriti i transitori di accensione il segnale inizia ad essere amplificato e quindi udibile negli altoparlanti.

Questa particolare sequenza di accensione è possibile per il fatto che la disattivazione delle funzioni Standby e Mute avviene per valori di tensione differenti e per l'esattezza la soglia di disattivazione del Mute è sopra i 2,9 volt mentre quella di accensione (ossia uscita dallo Standby) è di soli 1,3V. Ecco perché pur avendo collegato i due piedini alla stessa rete di temporizzazione si verifica prima l'uscita dallo standby e poi l'arrivo del segnale alle uscite. Secondo indicazioni del costruttore, un ritardo di $100 \div 200$ ms tra la disattivazione dello Standby e quella del Mute è idoneo al corretto funzionamento dell'amplificatore. Bene, concludiamo l'analisi del circuito con la linea di alimentazione, che fa capo alla morsettiere Vcc-GND alla quale si deve applicare una tensione continua di $16 \div 17$ V che viene filtrata dal condensatore elettrolitico C3; la linea positiva è protetta dall'inversione di polarità mediante il diodo D1.

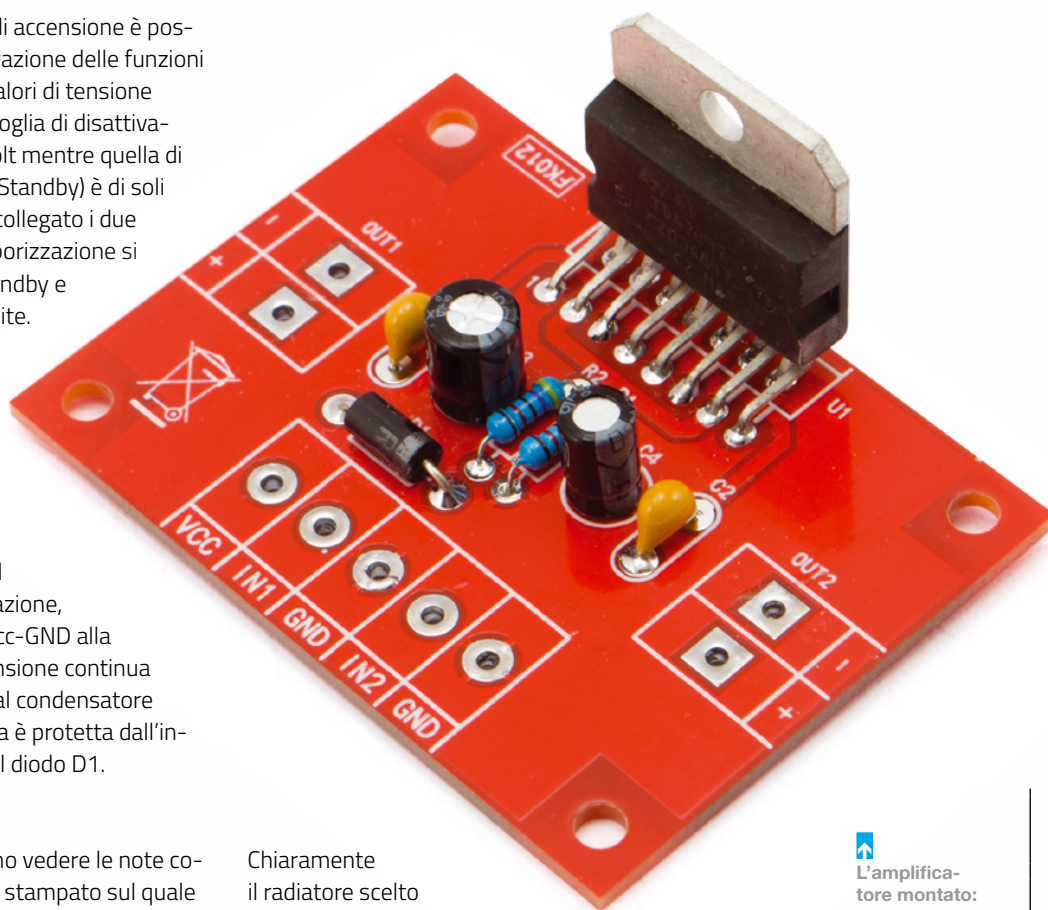
REALIZZAZIONE PRATICA

Giunti a questo punto possiamo vedere le note costruttive, partendo dal circuito stampato sul quale prenderanno posto i pochi componenti richiesti; il PCB è a doppia ramatura e dal nostro sito Internet <https://ei.futuranet.it> potete scaricare le due tracce da stampare su carta da lucido o acetato in modo da ottenere le pellicole per procedere con la fotoincisione.

Una volta inciso e forato il circuito stampato disponetevi i componenti, iniziando dalle resistenze e dal diodo (quest'ultimo ha un preciso orientamento che vedete nel piano di montaggio pubblicato in queste pagine) quindi procedendo con i condensatori, prestando attenzione alla polarità dei due elettrolitici.

Montate quindi le morsettiere per ingressi, uscite e alimentazione, quindi terminate con l'integrato TDA7297, che troverà posto nelle rispettive piazzole nelle quali entrerà nel verso giusto, perché è l'unico possibile.

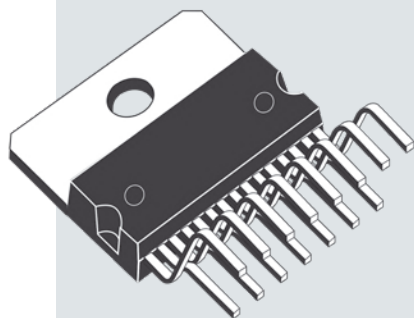
L'integrato dovrà essere dotato di un dissipatore di calore in alluminio da 2°C/W di resistenza termica da fissare al lato metallico mediante un bulloncino 3MA con dado, interponendo della pasta termica al silicone per agevolare lo smaltimento del calore che inevitabilmente verrà prodotto durante il funzionamento.



Chiaramente il radiatore scelto dovrà avere una superficie liscia abbastanza ampia da ospitare il corpo dell'integrato, che è in package Multiwatt 15V; l'ideale sarebbe un dissipatore con una faccia piatta, così da facilitare la dissipazione del calore quando il circuito, come è verosimile, verrà inserito in un contenitore per elettronica. Completato il montaggio e verificatane l'esattezza, il circuito è pronto per l'uso, in quanto non richiede alcuna operazione di messa a punto o taratura preliminare.

Per funzionare, il circuito richiede una fonte di alimentazione capace di erogare $16 \div 17$ V (ma anche 18 V) in continua e una corrente di circa 2,5 A; va bene, per esempio, un trasformatore da rete con secondario da $12 \div 13$ Vca – 2,5 ampere, seguito dal classico ponte raddrizzatore 60V-4A (un KBL04 andrà benissimo) e da un condensatore elettrolitico di livellamento da $4.700 \mu\text{F} / 25\text{V}$. Ricordiamo che il nostro amplificatore è utilizzabile anche come booster, tenendo però presente che nelle automobili moderne la tensione dell'impianto elettrico non è stabile al valore massimo erogabile dall'alternatore, bensì può variare tra meno di 12V e oltre 14V in base alle impostazioni di funziona-

 L'amplificatore montato: al TDA7297 va applicato il dissipatore.



Il componente chiave dell'amplificatore descritto in questo articolo appartiene alla fortunata famiglia di finali single-chip a due stadi di potenza a ponte, capostipiti dei quali sono stati il TDA2004 e il TDA2005.

Il più moderno

TDA7297 contiene due amplificatori indipendenti con uscita a ponte, operanti in classe AB, molto facili da utilizzare perché integrano le reti di polarizzazione e retroazione, nonché la connessione per lo sfasamento dei segnali che pilotano i due finali. L'integrato è quindi adatto a realizzare, con l'ausilio di pochissimi componenti esterni, un amplificatore stereo.

Le caratteristiche tecniche sono di tutto rispetto e designano il TDA7297 quale amplificatore finale di impianti hi-fi e riproduttori musicali compresi quelli di file mp3; la distorsione armonica

(THD) è di appena lo 0,025% @1 kHz a una potenza d'uscita di $2 \times 3W$ su 8 ohm a 16,5Vcc di alimentazione, che sale al 10% alla massima potenza, indipendentemente dalla frequenza del segnale.

La separazione tra i canali è di 60 dB, quindi più che accettabile e la risposta in frequenza a - 3dB, 1W su 8 ohm a 16,5Vcc di alimentazione spazia fra 20 e 100 kHz.

Il guadagno in tensione è fissato internamente in 40 volte e non può essere modificato dall'esterno perché le reti di retroazione degli operazionali di potenza sono interne.

L'integrato è dotato di protezione da sovratemperatura e da sovraccarico in uscita (interviene a 2 ampere, corrente ben al di sopra di quella richiesta per sviluppare la massima potenza su 8 ohm) compreso il cortocircuito; dispone anche di funzionalità di Mute e Standby, attivabili dai rispettivi piedini di controllo allo scopo di poter spegnere o accendere con una logica (un micro-controllore, per esempio) gli stadi di potenza pur mantenendo collegata l'alimentazione, come nel caso degli apparati funzionanti a batterie ma anche di garantire accensioni e spegnimenti senza i classici rumori tipici dei transistori.

mento fatte dalla ECU dietro una serie di valutazioni sullo stato di carica della batteria e la potenza richiesta dai vari utilizzatori. Questo implica che alla massima potenza possono esserci distorsioni quando la tensione scende.

Qualunque sia l'impiego, rammentate che il dissipatore di calore dell'integrato si trova a potenziale di massa, quindi va sempre mantenuto isolato dal contenitore in cui racchiuderete l'amplificatore, se di metallo; potete isolare il dissipatore interponendo un foglietto di mica isolante tra l'aletta metallica del TDA7297 e il corpo, spalmando entrambe le facce con del grasso al silicone o altra forma di pasta termica.

A proposito di contenitore e di inserimento in esso, l'ideale sarebbe poter montare il circuito in modo che il dissipatore si affacci all'esterno dalla parte posteriore, in modo da ottenere uno smaltimento

del calore ottimale. Ma nulla vieta di adottare un contenitore dove il circuito verrà racchiuso totalmente, solo che in tal caso sarà opportuno prevedere delle griglie o feritoie sotto e sopra al dissipatore, così da creare un flusso d'aria (convezione) in grado di asportare il calore prodotto, specie alla massima potenza.

Agli ingressi potete collegare due spezzoni di cavetto schermato coassiale audio terminanti ciascuno su una presa RCA, avendo cura di saldare il filo centrale (anima o conduttore caldo, che dir si voglia...) al contatto interno della presa e la calza schermo a quello esterno. Potete tranquillamente utilizzare le prese RCA stereo montate su flangia da montare a vite, così da ottenere un insieme più compatto.

Per collegare gli altoparlanti potete ricorrere ai classici morsetti a molla in uso in buona parte degli apparati hi-fi, connettendo i due fili di ogni uscita ai terminali dei morsetti; ricordiamo ancora una volta che le uscite del TDA7297 sono sollevate da massa, pertanto nessuno dei due fili o morsetti dovrà entrare in contatto con la massa; in vero, se collegate l'altoparlante tra un'uscita e massa il circuito funzionerà ugualmente ma svilupperà una potenza in altoparlante di appena 1/4 di quella nominale.

Bene, con questo abbiamo concluso: ritenendo di avervi dato tutte le indicazioni per una realizzazione di successo, non ci resta che augurarvi buon ascolto!



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
FK012	Amplificatore audio stereo 2x15 watt	8,50

IL MATERIALE VA RICHIESTO A:

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

Indispensabili in ogni laboratorio!

STAZIONE SALDANTE 60 W
CON BRACCI FLESSIBILI
E TEMPERATURA
REGOLABILE
DA 160 A 480°C

cod. SS203
€39,00



MINI STAZIONE SALDANTE 10W
CON TEMPERATURA
REGOLABILE
DA 100 A 450°C
IDEALE PER
COMPONENTI SMD

cod. SS202N
€19,90



STAZIONE SALDANTE 130 W
CON TEMPERATURA
REGOLABILE
DA 160 A 480°C
STAGNO,
PORTA STAGNO,
PAGLIETTA
IN METALLO
E SPUGNA

cod. TP237
€49,00



STAZIONE SALDANTE 48 W
CON DISPLAY LCD E
INDICAZIONE DELLA
TEMPERATURA IMPOSTATA E
DELLA TEMPERATURA REALE
DA 150 A 450°C

cod. VTSSC40N
€94,00



STAZIONE SALDANTE 48 W
CON TEMPERATURA
REGOLABILE DA 150 A 480°C

cod. TPGSS501
€21,00



STAZIONE SALDANTE/
DISSALDANTE PER COMPONENTI
SMD CON SALDATORE
DA 160 A 480 °C E
STILO AD ARIA CALDA
DA 160 A 480 °C

cod. TP282
€278,00



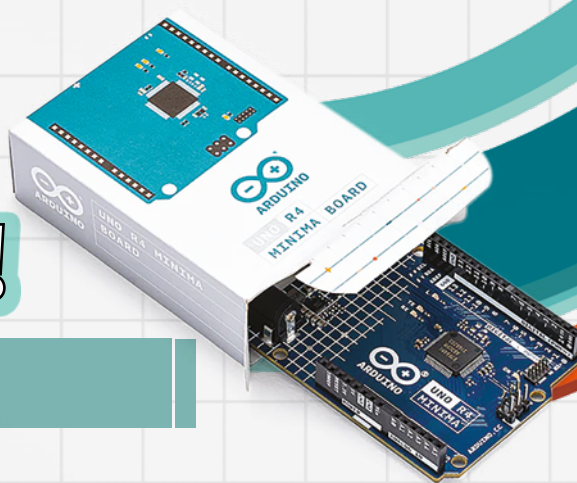
STAZIONE SALDANTE/
DISSALDANTE PER COMPONENTI
SMD CON SALDATORE
DA 50 A 480 °C
E STILO AD ARIA CALDA
DA 100 A 500 °C

cod. VTSS230
€165,00



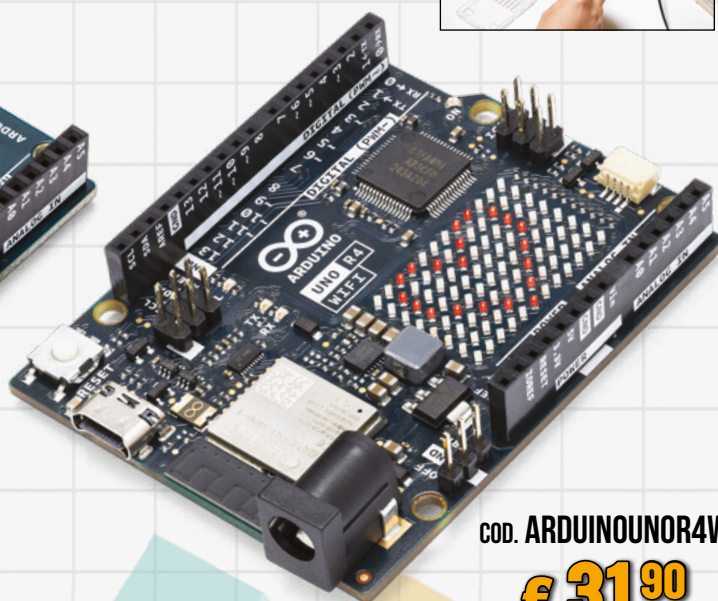
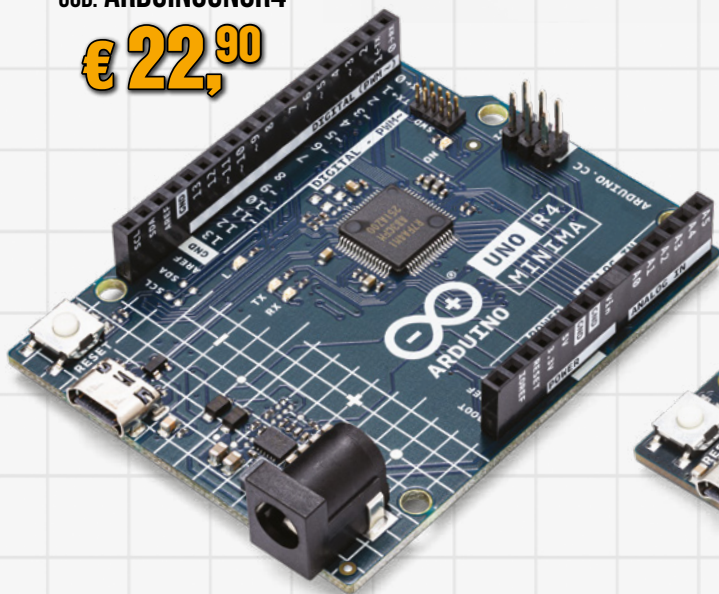
Scopri la gamma completa su **www.futurashop.it**

Scopri Arduino UNO R4!!!



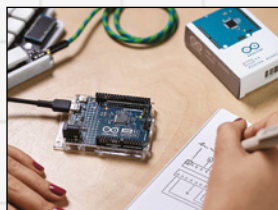
COD. ARDUINOUNOR4

€ 22,90



COD. ARDUINOUNOR4W

€ 31,90



Prezzi IVA inclusa.

CARATTERISTICHE TECNICHE		UNO R4 MINIMA	UNO R4 WiFi
MCU	R7FA4M1AB3CFM#AA0 (Arm cortex M4)	✓	✓
Memoria	256KB Flash / 32KB SRAM	✓	✓
Tensione operativa	5V	✓	✓
Tensione d'ingresso	6-24V	✓	✓
Velocità Clock	48 MHz	✓	✓
Porta di programmazione	USB-C	✓	✓
WiFi/BLE	ESP32-S3-MINI		✓
LED Matrix	12x8 (96 LED rossi)		✓
Connessioni supplementari	-connettore Qwiic -pin OFF -pin VRTC		✓
Pin digitali	14	✓	✓
PWM	6	✓	✓
ADC	6	✓	✓
DAC	1 (12 bit)	✓	✓
SPI	1	✓	✓
I2C	1	✓	✓
CAN	1	✓	✓

SMART TARGET

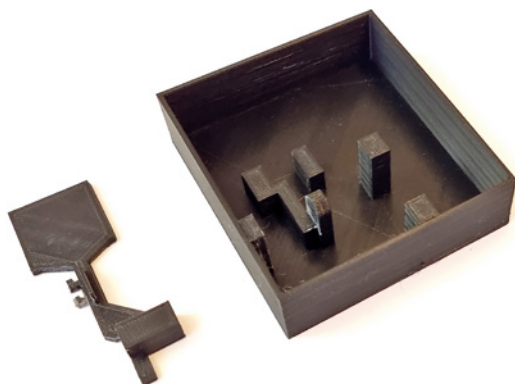
dell'ING. MIRCO SEGATELLO



Realizziamo bersagli avanzati, interconnessi in una rete, ideali per competizioni con pistole giocattolo. L'intero sistema è facilmente gestibile tramite smartphone, grazie all'integrazione del modulo ESP32.

Chi di voi non ha mai giocato con le pistole a pallini o i fucili giocattolo alzi la mano! Questi giochi sono molto popolari tra i bambini, forse meno tra le bambine, ma comunque riscuotono sempre molto successo, e se siete cresciuti, potete sempre passare al softair. Qualunque sia il vostro gioco preferito, l'obiettivo è sempre lo stesso: mirare e sparare. Vince chi ha la mira migliore. Anche in diverse fiere paesane si trova spesso uno stand con pistole ad aria compressa, il cui scopo è abbattere dei barattoli di latta, con la massima precisione possibile. Il gioco diventa interessante quando ci si sfida con gli amici a colpire più bersagli e scoprire chi ha la mira migliore, purtroppo la cosa noiosa è dover riposizionare i bersagli ogni

→ **Fig. 1**
Parti meccaniche
stampate in 3D.



→ **Fig. 2**
Parti meccaniche
assemblate con il
servo.



volta che vengono colpiti, inoltre non è sempre immediato determinare il vincitore. Ed è qui che entrano in funzione i nostri bersagli intelligenti, che permettono di rendere il gioco più dinamico e divertente.

Nell'ideare questo progetto, abbiamo anche preso spunto da alcune serie televisive poliziesche nelle quali si mostrava l'addestramento degli agenti di polizia tramite l'utilizzo di bersagli azionati auto-

maticamente. Nello sviluppare il progetto, abbiamo ipotizzato ci fossero tre bersagli che potessero essere azionati da remoto e il sistema dovesse tenere traccia del tempo impiegato per abbattere ciascun bersaglio. La possibilità di un azionamento a distanza ci permette di gestire anche diverse modalità di gioco, impossibili da realizzare con bersagli statici, ed è per questo che li chiamiamo bersagli intelligenti, in inglese smart target. Divideremo la presentazione del progetto in tre sezioni, la parte meccanica, elettrica ed informatica.

PARTE MECCANICA

Ci serve un bersaglio che possa essere azionato a comando, sia posizionato armato pronto per essere colpito, sia in disarmo in posizione di riposo, così facendo possiamo definire un tempo massimo entro il quale il bersaglio può essere colpito, dopodiché il bersaglio non sarà più disponibile e il tiratore sarà dichiarato perdente.

Il bersaglio potrà essere colpito in qualsiasi momento e dovrà posizionarsi in uno stato di off, senza che nessun organo meccanico venga guastato dal colpo dell'arma.

C'è voluto del tempo, ma alla fine abbiamo trovato la soluzione giusta, che ora andiamo a descrivere prendendo in considerazione le figure in questa pagina. Il sistema è composto da due parti stampate in 3D, una base ed un'asta, quest'ultima può ruotare liberamente attorno un perno ricavato da una semplice graffetta, per la realizzazione potete usare i file di stampa in formato STL disponibili assieme ai file di questo articolo. I pezzi stampati, visibili nella **Fig. 1**, vanno replicati tre volte uno per ciascun bersaglio.

L'asta può oscillare liberamente sul suo perno, mentre il movimento è affidato ad un servo da modellismo, come visibile nelle **Fig. 2 e 3**. Il servo utilizzato in questo progetto ha sigla SG90 ed è distribuito da Futura Elettronica con codice 7300-SERVO206, la sua particolarità è quella di poter ruotare di 180°, a differenza di altri che sono limitati a circa 120°. La molla che completa il movimento, visibile nella **Fig. 2**, è stata recuperata da un vecchio giocattolo, in alternativa anche un elastico può andare bene.

Inizialmente pensavamo di fissare l'asta del bersaglio direttamente al perno del servo da modellismo, in modo da permetterne il movimento diretto, ma così facendo in seguito ad un colpo il bersaglio non sarebbe caduto, anzi si sarebbero rotti gli ingranaggi del servo, cosa assolutamente da evitare. Nel progetto definitivo l'asta porta bersaglio, pur essendo

→ **Fig. 3**
Il servo da
modellismo.



in asse con il perno del servo, è disgiunta meccanicamente dal servo e connessa tramite una molla. Per capire come funziona il meccanismo, riferiamoci alla **Fig. 4** nella quale abbiamo rappresentato in modo semplificato i componenti coinvolti. Nella sezione A, vedete il bersaglio in posizione OFF e il servo posizionato a 180°, in questa situazione la molla (rappresentata dalla linea nera) porterà il bersaglio in posizione verticale (ON), rendendolo pronto ad essere colpito. Da questa posizione il servo si porterà a 90° (sezione B) e grazie alla molla verrà applicata una debole forza per farlo rimanere in verticale. Da questa posizione, quando il bersaglio verrà colpito, l'asta si abbasserà per riportarsi in posizione di OFF, la molla garantirà una leggera forza di trattenimento verso il basso, garantendo che il bersaglio non rimbalzi e si riporti in posizione verticale (sezione C). Sempre partendo dalla posizione ON, posizionando il servo a 0° si abbasserà il bersaglio (sezione D), successivamente, il servo viene portato nella condizione di riposo a 90° assicurando comunque che venga mantenuta la posizione. Con il servo in posizione di riposo a 90°, sarà anche possibile posizionare manualmente il bersaglio; in ogni caso, non vi sarà alcun danno alla meccanica. Il dettaglio del funzionamento di questo meccanismo è visibile nel video, il cui link è disponibile nell'apposito riquadro dell'articolo.

CABLAGGIO

Pochi altri componenti sono necessari per completare il progetto, uno tra questi è il sensore che rileva l'abbattimento del bersaglio, si tratta di un finecorsa ottico utilizzato nelle stampanti 3D distribuito da Futura Elettronica, con codice 2846-ENDSTOP3D e visibile in dettaglio nella **Fig. 5**. Inizialmente la soluzione era di utilizzare un più comune finecorsa meccanico ma, il violento impatto del proiettile, comportava un forte colpo sul finecorsa pregiudicandone la durata nel tempo; nel caso di un sensore ottico questo problema non si presenta. Questo sensore dispone di soli tre contatti, l'alimentazione positivo e negativo e il segnale di uscita, facilmente interfacciabile con una logica programmabile. Abbiamo previsto la presenza di tre bersagli distinti, che possono essere posizionati a piacimento, ma con la possibilità di essere gestiti a distanza in modalità wireless, senza la necessità di collegamenti elettrici che avrebbero reso difficoltosa l'installazione. Serve pertanto una logica programmabile con capacità di comunicazione senza fili, facilmen-

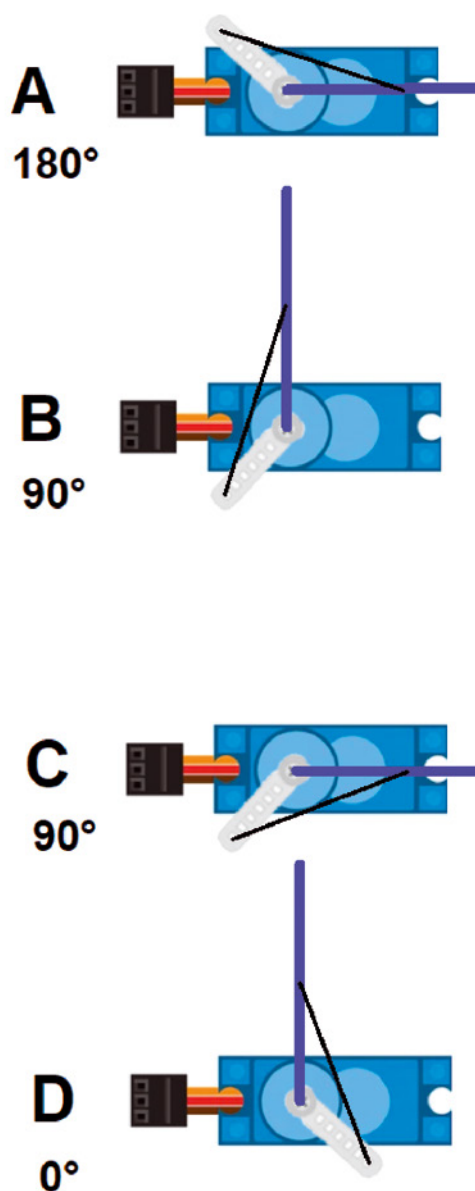


Fig. 4
Azionamento
meccanico
del bersaglio.

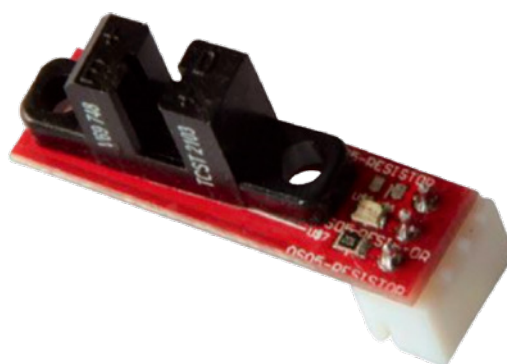
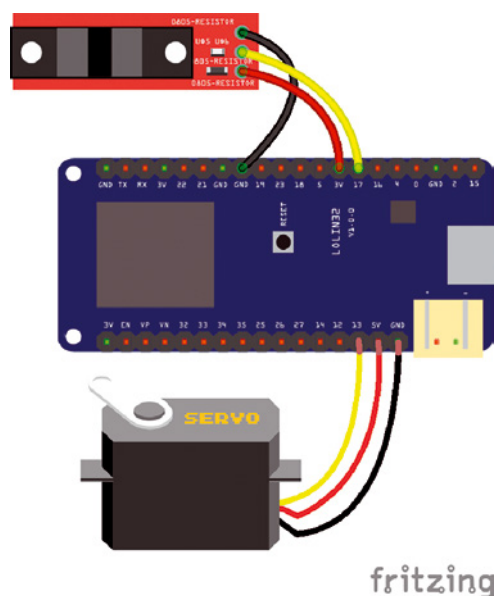


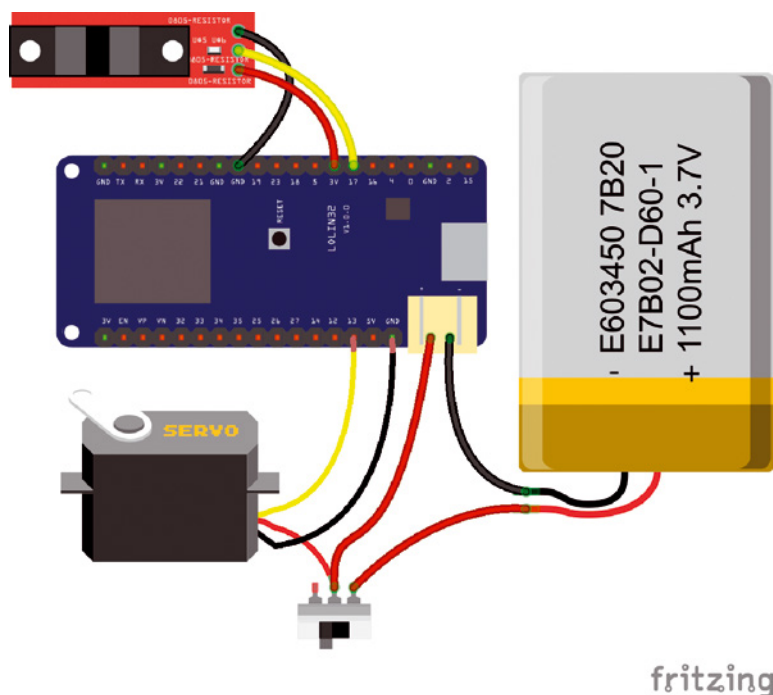
Fig. 5
Sensore
rilevamento
abbattimento.

Fig. 6
Collegamenti elettrici con alimentazione da USB.



te integrabile con un'alimentazione indipendente a batteria. La scelta è caduta sulla scheda ESP32 nella versione WEMOS LOLIN32, distribuita con codice 8300-WEMOSL32. Si tratta di una board equipaggiata con un modulo ESP32, che integra al suo interno un modulo BT e WiFi, inoltre la scheda è predisposta per essere alimentata da una batteria agli ioni di litio ricaricabile grazie alla presenza di

Fig. 7
Collegamenti elettrici con alimentazione a batteria.



un chip di ricarica integrato. Qualsiasi altra scheda basata su modulo ESP32 è comunque adatta per il nostro progetto. Per quanto riguarda i collegamenti elettrici abbiamo previsto due varianti a seconda della fonte di alimentazione che decidiate di utilizzare. Nel primo caso la scheda viene alimentata con un semplice caricabatteria da smartphone con uscita a 5V e spina micro USB; non è necessario alcun interruttore di accensione. I collegamenti sono visibili in **Fig. 6**. Il servo può essere inserito direttamente con la sua spina in quanto i suoi pin coincidono perfettamente con la scheda, mentre il finecorsa ottico sarà collegato alla board con il cavetto in dotazione.

Nel secondo caso la scheda viene alimentata da una singola cella LiPo collegata all'apposito connettore. In questo caso è necessario prevedere un interruttore di accensione. Il servo da modellismo dovrà prelevare la tensione di alimentazione direttamente dalla batteria (come visibile in **Fig. 7**), non dal pin 5V che è privo di tensione e neppure dal pin 3V3, che non fornisce sufficiente tensione.

La batteria ai polimeri di litio 3,7 V 1000 mA con connettore 2mm (codice 6374-PLIB1000-2MM), trova posto all'interno del bersaglio, come visibile in **Fig. 8**, mentre il finecorsa ottico viene collegato come nello schema precedente. Tutti i cavi possono essere ripiegati all'interno della base in modo da non farli interferire con il meccanismo. Per concludere, un suggerimento pratico: essendo il bersaglio leggero potrebbe spostarsi qualora venga colpita la base invece del bersaglio, in tal caso conviene riempire il contenitore con del materiale pesante, come dei pesetti di piombo utilizzati dai pescatori.

In **Fig. 9** potete vedere i tre bersagli assemblati, notate come alle palette sono stati fissati dei cartoncini raffiguranti il bersaglio vero e proprio. Potete sbizzarrirvi a vostro piacimento, il nostro è solo un piccolo esempio.

IL CODICE

Per lo sviluppo del software dobbiamo considerare tre punti fondamentali: il primo riguarda la gestione del bersaglio, il secondo è la comunicazione tra i bersagli, e il terzo punto è l'interfaccia utente. Per la gestione del servo da modellismo ci siamo avvalsi della libreria di nome ESP32servo reperibile all'indirizzo <https://github.com/jkb-git/ESP32Servo>. Nei file dell'articolo trovate oltre alla libreria anche un esempio di nome servo_test_ESP.ino utile per testare e tarare il servo e la funzionalità meccanica in generale. Quando il perno del servo si trova

nella posizione centrale a 0°, la squadretta dovrà risultare a 45° rispetto la base, come visibile nella **Fig. 2**. Collegate un potenziometro tra i 3V3 e GND e il cursore al pin 36, e avviate lo sketch indicato in precedenza; potrete regolare manualmente la posizione del servo, e su Serial Monitor sarà visualizzato il valore in microsecondi corrispondente alla posizione assunta dalla squadretta. Ruotate il potenziometro finché su Serial Monitor comparirà il valore di 1500, corrispondente al valore centrale del perno, in questa situazione la squadretta dovrà risultare posizionata a 45° rispetto la base (punti B e C della **Fig. 4**), se così non fosse, riposizionate la squadretta nel perno. Successivamente posizionate la squadretta del servo in corrispondenza dei punti A e D della **Fig. 4**, questi due valori (espressi in microsecondi) sono i due valori estremi di rotazione del perno del servo, e corrispondono ai valori posMIN e posMAX riportati nel software (si veda il **Listato 1**). Se utilizzate il servo SG90, come nei nostri prototipi, i due valori di default dovrebbero andare bene. A livello software sono stati definiti tre punti in cui verrà posizionato il servo e sono rispettivamente posON, qualora il bersaglio venga posizionato pronto per essere colpito, posOFF qualora il bersaglio venga posizionato in basso e per finire posMID che corrisponde alla posizione

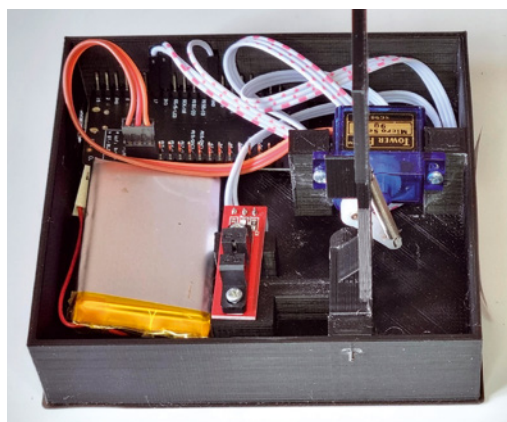


Fig. 8
Interno del bersaglio con installata la batteria.

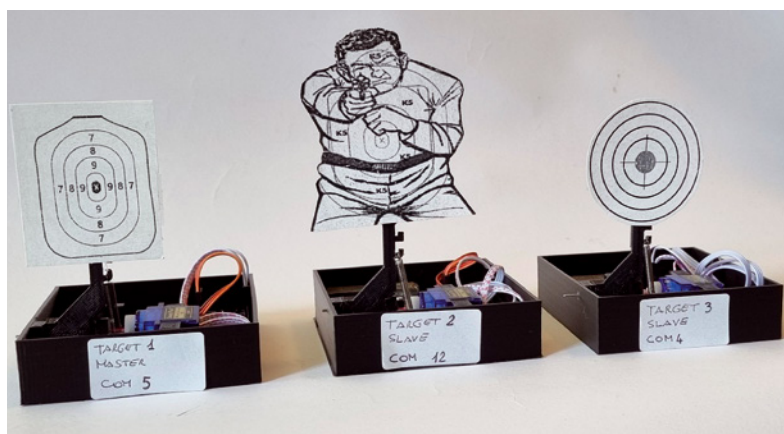


Fig. 9
Bersagli assemblati.

Listato 1 - Definizione servo e pin

```
int posMAX = 2400;
int posMIN = 500;
int posON = 180;    // high target
int posMID = 90;    // central target
int posOFF = 0;     // low target
int servoTime = 600; // time for move in position
const int ledPin = 5;
const int sensorPin = 17; // optical sensor pin
const int servoPin = 13; // servo pin
```

Listato 2 - Azionamento servo

```
void target1ON()
{
  Serial.println("Target1 ON");
  myservo.write(posON);
  delay(servoTime);
  myservo.write(posMID);
  target1State=targetON;
}
void target1OFF()
{
  Serial.println("Target1 OFF");
  myservo.write(posOFF);
  delay(servoTime);
  myservo.write(posMID);
  target1State=targetOFF;
}
```

di riposo, nella quale il bersaglio può assumere entrambe le posizioni ON e OFF. La libreria farà corrispondere automaticamente i gradi rotazione 0° e 180° ai due valori posMIN e posMAX indicati. Quando il bersaglio è abbassato e vogliamo portarlo in posizione di tiro, sarà sufficiente dare il comando `myservo.write(posON)` e l'asta con il bersaglio verrà portata in posizione, dopo un piccolo ritardo il servo viene portato in posizione di riposo con il comando `myservo.write(posMID)`, come riportato nel **Listato 2**.

Sempre facendo riferimento al **Listato 1**, le costanti servoPin e sensorPin definiscono rispettivamente a quale pin è connesso il servo e a quale pin è connesso il sensore ottico. Quest'ultimo ci fornirà l'informazione fondamentale di quando il bersaglio viene abbattuto, calcolando la differenza di tempo tra il momento in cui il bersaglio è a tiro e quando il sensore darà segnale, sapremo in quanto tempo il bersaglio è stato colpito, dato fondamentale per capire chi è il più bravo nel gioco. Quanto riportato nel **Listato 1 e 2** è comune al firmware di tutti i

I microcontrollori ESP32 ed ESP8266 supportano comunicazioni wireless Bluetooth e WiFi. La parte radio può però gestire altri protocolli di comunicazione nella banda ISM a 2,4 GHz. Espressif, produttore dei moduli ESP, ha sviluppato il protocollo ESP-NOW, che crea una rete wireless privata con ricetrasmittitori a 2,4 GHz. Questa rete coesiste con il Wi-Fi ed è utilizzabile solo da microcontrollori ESP, permettendo l'invio di brevi pacchetti di dati senza necessità di connessione.

Ogni messaggio tra dispositivi può contenere al massimo 250 byte. Il protocollo è versatile, consentendo comunicazioni sia unidirezionali che bidirezionali. I dispositivi possono

fungere sia da master che da slave, permettendo a qualunque dispositivo di inviare o ricevere dati. I pacchetti sono indirizzati tramite l'indirizzo MAC, a differenza delle reti LAN che usano indirizzi IP. È possibile inviare dati a tutti i dispositivi contemporaneamente in modalità broadcasting. Prima di comunicare, è necessario accoppiare i dispositivi. La connessione è sicura e l'accoppiamento è persistente, anche dopo un riavvio. La limitazione di 250 byte rende questo protocollo adatto per comandi remoti ad attuatori o lettura di dati da sensori. La portata può arrivare a 400 metri in condizioni ottimali, sufficiente per un'abitazione, con un massimo di 20 dispositivi per rete.

bersagli; è però necessario definire un bersaglio master che sovrintende tutte le operazioni e gestirà l'interfaccia utente, mentre gli altri target saranno semplicemente degli slave pronti ad eseguire i comandi impartiti dal master, il tutto secondo lo schema riportato in **Fig. 10**.

Guarda lo Smart Target in azione su YouTube!



Vuoi vedere in azione il nostro Smart Target? Nulla di più semplice: collegati alla pagina:
<https://youtu.be/OM2kK9B47VE>
Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.

Il master dovrà integrare al suo interno anche un web server che ci consentirà di realizzare l'interfaccia utente con la quale potremmo impostare il tipo di gioco e visualizzare il punteggio. Abbiamo previsto due tipi di gioco che abbiamo chiamato SPEED e RANDOM. Nel primo, tutti i bersagli vengono posizionati su ON e compito del giocatore è quello di abatterli nel minor tempo possibile. Nell'interfaccia utente, oltre a poter far partire il gioco dovranno potersi visualizzare i tempi realizzati per colpire ogni singolo bersaglio, più il tempo totale necessario ad abatterli tutti; ovviamente il vincitore sarà quello con il tempo più basso. Nel secondo gioco invece, i bersagli sono posizionati uno alla volta in modo casuale su ON e il tiratore dovrà colpirli entro un tempo massimo prestabilito, superato questo tempo il bersaglio si disabilita portandosi in OFF e se ne attiverà un altro, sempre casualmente. Il punteggio viene calcolato sommando il numero dei bersagli colpiti nel tempo massimo prestabilito.

Per realizzare il web server ci siamo appoggiati alla libreria di nome ESPDASH scaricabile all'indirizzo <https://github.com/ayushsharma82/ESP-DASH>. Questa libreria consente di realizzare web server interattivi in modo semplice e veloce con una grafica moderna ed accattivante senza dover scrivere neppure una riga di codice html; la versione free è già sufficiente per realizzare discrete funzionalità. Nelle **Fig. 11** e **12** sono riportate le schermate nel caso dei due giochi menzionati. Il web server funziona in tempo reale quindi i tempi saranno aggiornati non appena il bersaglio verrà colpito, ovviamente se non verrà colpito nel tempo massimo prestabilito si disabilita in automatico e il giocatore sarà dichiarato perdente.

Nel listato del master trovate le impostazioni relative ai tempi di gioco: targetTimeMax e targetRandomMax. Il primo tempo è relativo al massimo tempo che il giocatore può utilizzare per abbattere tutti i bersagli nella modalità SPEED. Il secondo tempo è invece la durata del gioco in modalità RANDOM, durante la quale i bersagli si azionano in sequenza casuale.

Il software è stato sviluppato come una macchina a stati, una per ciascun bersaglio, in modo da tenere traccia della sequenza di gioco, in ogni situazione è previsto un timeout che disattiva il bersaglio se non viene colpito entro un certo tempo; funzionalità impossibile con i bersagli manuali. In ogni momento, possiamo anche azzerare il gioco disabilitando tutti i bersagli con un solo comando da remoto.

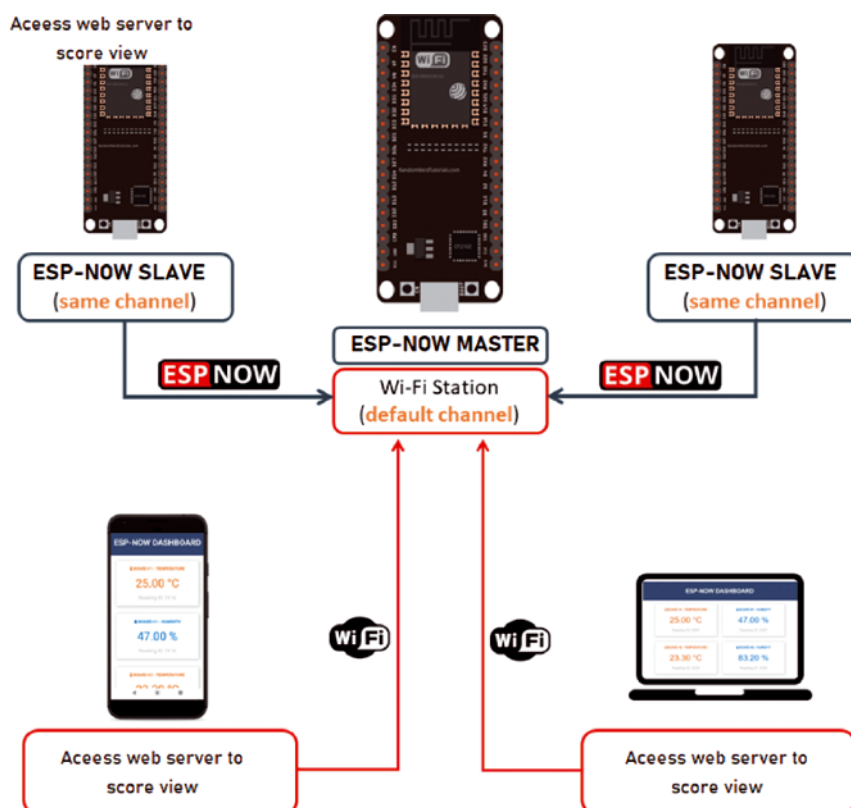


Fig.10
Configurazione della rete dei bersagli.

INSTALLAZIONE DEL FIRMWARE

Disponendo di tre moduli ESP32, dovreste fare molta attenzione all'installazione del firmware, daremo per scontato che abbiate già avviato l'IDE di Arduino, configurato per gestire la programmazione dei moduli ESP32 e aperto lo sketch di nome SmartTargetMaster.ino; nella sezione strumenti dell'IDE impostate come scheda la versione WEMOS LOLIN32. Prima di caricare lo sketch dovete necessariamente impostare alcuni parametri relativi al funzionamento del servo come descritto precedentemente, successivamente dovete impostare la modalità di accesso al web server, ci sono due possibilità: la prima prevede di usare la rete di casa esistente (STA mode), nel caso dovreste inserire il nome e la password di accesso, come visibile nel listato 3, nel secondo caso (AP mode) sarà il modulo ESP a creare una rete privata. Quest'ultimo caso è necessario in assenza di una rete preesistente, come nel caso di un gioco che si sviluppa in giardino. Il dispositivo che utilizzerete per accedere al webserver dovrà agganciarsi alla rete Wi-Fi creata dal modulo ESP, per farlo dovreste andare nelle impostazioni di rete e selezionare quella di nome SMART TARGET, dopodiché, per aprire il web server, aprite il browser e inserite l'indirizzo 192.168.1.4. Se utilizzerete la modalità STA,

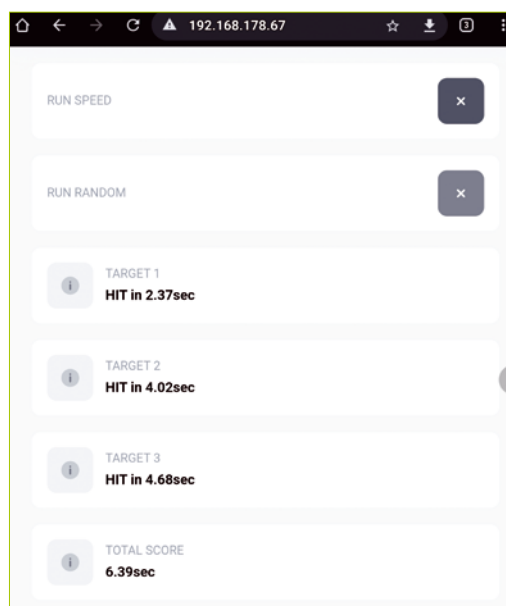


Fig.11
Web Server per SPEED game.

una volta completata la programmazione, aprite il Serial Monitor di Arduino per visualizzare l'indirizzo IP assegnato alla scheda (**fig. 13**), quest'ultimo andrà inserito nel browser del dispositivo, smartphone o tablet, che utilizzerete per gestire il web server. Successivamente, aprire lo sketch di nome Smart-

Fig. 12
Web Server per
RANDOM game.

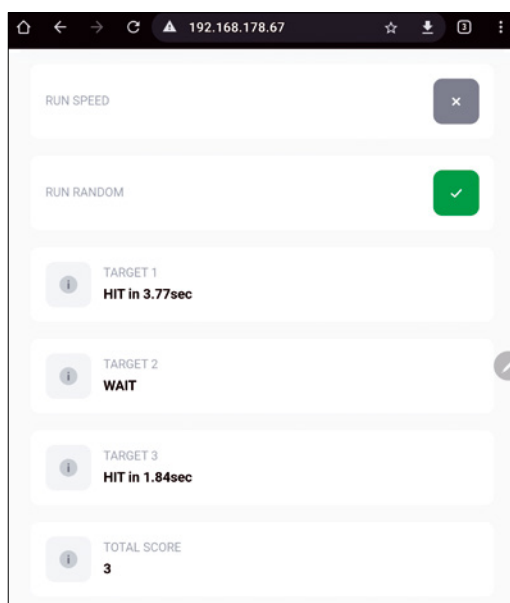
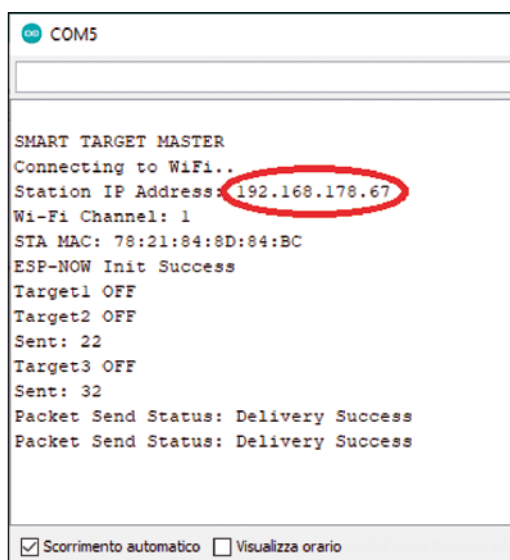


Fig. 13
Serial Monitor in
modalità STA.



Listato 3 - Impostazioni di rete

```
#define STA_mode
//define AP_mode
// Credentials for network (AP mode)
const char* AP_ssid = "SMART TARGET"; //HS Name
const char* AP_password = "12345678"; //HS PWD
// Credentials for home network (STA mode)
const char* STA_ssid = "SSID"; //Router name
const char* STA_password = "PSW"; //Router PWD
```

TargetSlave.ino e settate correttamente i pin di collegamento e i parametri di funzionamento del servo se diversi dalla configurazione proposta. Per evitare di dover configurare gli indirizzi MAC dei dispositivi presenti nella rete, le comunicazioni avvengono in broadcast mode, inviando sempre i messaggi a tutti i dispositivi indistintamente. Per capire da quale dispositivo arriva il dato utile abbiamo utilizzato dei semplici codici in quanto gli unici comandi che dovrà impartire il master verso gli slave è posizionare il targetON oppure targetOFF. Lo slave invia un dato solo quando il suo bersaglio viene colpito, sarà comunque il master a calcolare i tempi. Nello sketch il primo slave sarà impostato come di seguito:

```
#define commandTX 20 //command send target HIT
#define commandON 21 //command received target ON
#define commandOFF 22 //command received target OFF
```

Mentre nel secondo slave dovreste impostare i valori iniziando con il numero 3 (quindi 30, 31 e 32). Questi codici sono replicati nel master in modo da poter interagire con gli slave. Con un pizzico di programmazione si potrà espandere il sistema sino a circa una ventina di dispositivi. A configurazione conclusa, procedete alla programmazione. Aprendo il Serial Monitor avrete un report delle comunicazioni con il master, utile per fare una diagnostica del sistema.

CONCLUSIONI

Abbiamo effettuato diverse prove con pistole giocattolo, e la meccanica ha retto senza problemi. Usando la versione gratuita della libreria ESPDASH, non è stato possibile inserire nel web server alcuni campi per l'impostazione dei tempi di gioco, operazione che richiede l'editing del software. Nonostante alcune limitazioni, il gioco è estremamente fruibile e i tester si sono divertiti, trovando la sfida a due particolarmente soddisfacente. Anche giocando da soli, ci sono molte soddisfazioni, con l'obiettivo di migliorare i tempi di abbattimento. □



COSA OCCORRE ?

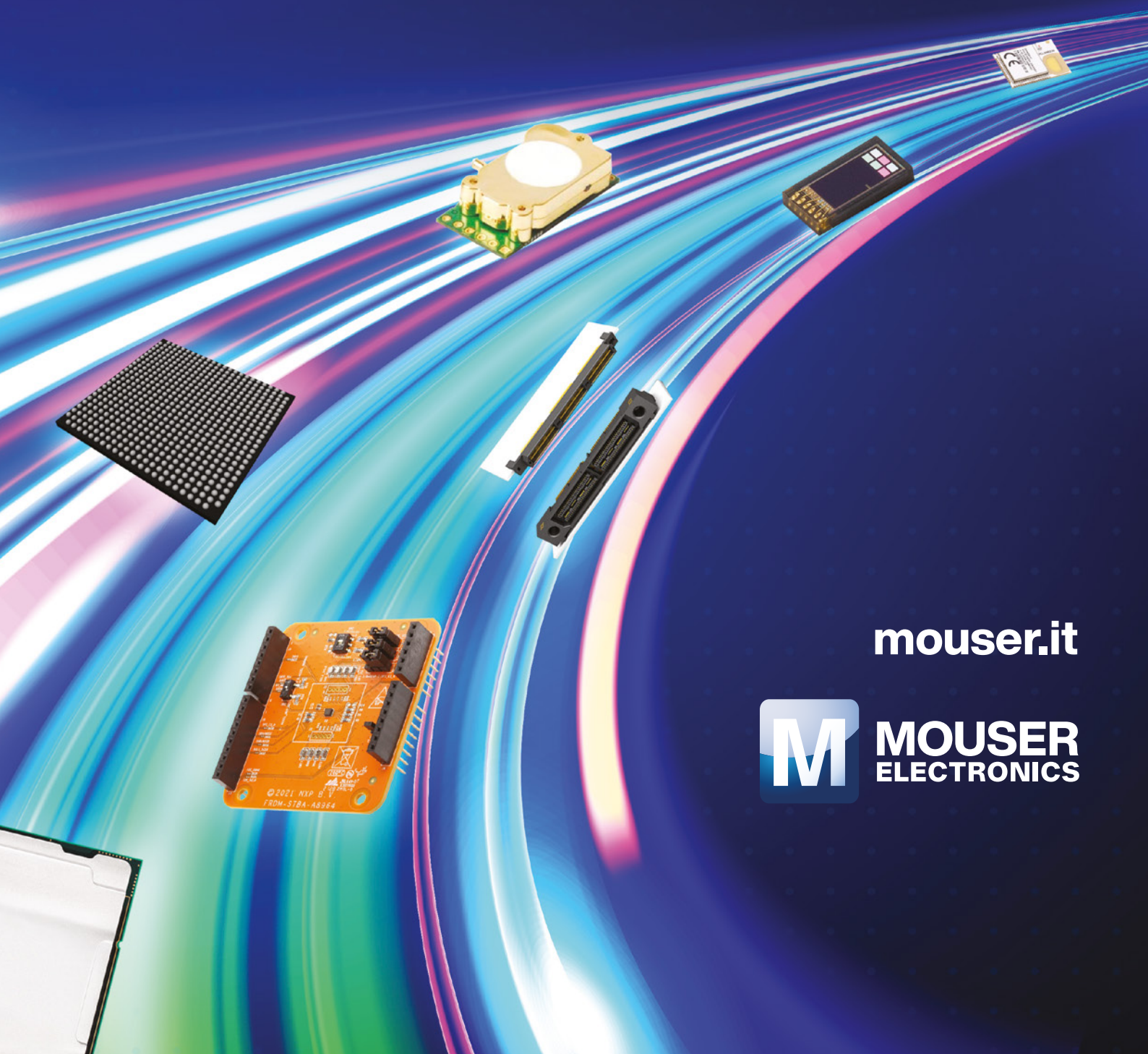
Codice	Descrizione	Prezzo € IVA inclusa
WEMOSL32	Board con ESP32 e caricabatteria	16,00
SERVO206	Servo motore 1.2kg/cm - 180°	7,50
PLIB1000-2MM	Batteria ai polimeri di litio 3,7 V 1000 mA	6,90
ENDSTOP3D	Finecorsa Ottico	3,50

IL MATERIALE VA RICHIESTO A:

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

A tutta velocità

Affidatevi al leader nell'introduzione di nuovi prodotti™ per passare dall'idea al prototipo alla velocità della luce



mouser.it



**MOUSER
ELECTRONICS**

Oscilloscopio palmare 1 canale - 200 KHz



Prezzo IVA inclusa.

€ 79,00

cod. DSO200

Dimensioni compatte, solo 99 x 69 x 20 mm! Dispone di display LCD a colori da 2,8 pollici con risoluzione di 320 x 240 pixel. Il DSO200 è alimentato da una batteria al litio integrata da 1000 mAh, ricaricabile tramite la porta USB di tipo C (cavo incluso). Il supporto ribaltabile a 90 gradi facilita l'uso su superfici piane. Sonde e accessori inclusi nella confezione.

- › Frequenza di campionamento: 2,5 Msa/s (milioni di campioni al secondo)
- › Sensibilità verticale: da 10 mV/Div a 20 V/Div
- › Base dei tempi: da 10 μ s/Div a 50 s/Div
- › Range tensione di test:
X1: 80 Vpp (-40 a 40 V)
X10: 800 Vpp (-400 a 400 V)
- › Accoppiamento: AC e DC
- › Calibrazione (onda quadra):
Frequenza: 1 kHz
Ciclo di lavoro: 50%
- › Tipo di trigger: fronte in salita o discesa
- › Modalità di trigger: automatica, normale, singola misurazione
- › Peso: 100 grammi

PROGRAMMAZIONE MICROCONTROLLORI ARM

di MAURIZIO ŠKERLIČ

Esploriamo metodi per programmare microcontrollori ARM, utilizzando strumenti economici e adattamenti pratici, rendendoli accessibili agli appassionati di elettronica hobbistica.

S

e con l'ambiente di sviluppo Arduino IDE scrivere, compilare e caricare i sorgenti nei microcontrollori è diventato facilissimo non si può dire la stessa cosa quando bisogna caricare il firmware nella Flash dei microcontrollori con architettura ARM a 32 bit vergini, cioè privi di bootloader.

Ogni famiglia di microcontrollori richiede un programmatore specifico, incompatibile con altre famiglie. In commercio tuttavia si trovano programmatori universali in grado di programmare la maggior parte dei microcontrollori disponibili, però sono costosi. In questo articolo vogliamo aiutare l'hobbysta nella programmazione di questi dispositivi utilizzando dei programmatori economici e a costruirne uno a basso costo utile per la programmazione dei microcontrollori della famiglia STM32.

I MICROCONTROLLORI ARM

Tutti conosciamo i microcontrollori ARM o più precisamente l'architettura ARM sviluppata dall'azienda *ARM holding* precedentemente conosciuta con il nome di *Advanced RISC Machine*. Indica una famiglia di microprocessori RISC (*reduction instruction set computer*) ossia computer a set di istruzioni ridotte a 32 bit e a 64 bit utilizzata in una moltitudine di sistemi e dispositivi embedded (*dedicati*).

Se negli ultimi anni le case produttrici di semiconduttori AMD e Intel hanno focalizzato i loro sforzi nello sviluppo di CPU sempre più performanti con prestazioni sempre migliori, ARM e i suoi partner si sono focalizzati nella loro integrazione e nella riduzione dei consumi. Il risultato sono le CPU della serie ARM11 e successivamente Cortex-A che hanno iniziato la grande rivoluzione e continuano ancora oggi ad essere implementati nella maggior parte degli smartphone. Con la speranza che la stessa storia si possa ripetere, ARM holding oggi si è focalizzata sulla progettazione di una nuova serie di CPU, con dimensioni ancora più ridotte e un consumo ancora minore, conosciuta con il nome di Cortex-M.

Tre sono le principali serie della famiglia di CPU ARM Cortex, la serie **Cortex-A** utilizzata per processori applicativi, la serie **Cortex-R** utilizzata per processori in tempo reale e la serie **Cortex-M** utilizzata per microcontrollori embedded.

Sebbene Cortex-A è la serie di CPU più conosciuta

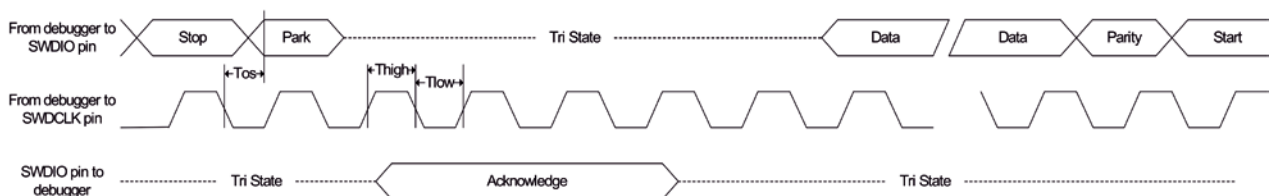
poiché utilizzata spesso negli smartphone, la serie Cortex-M sta diventando sempre più importante in quanto utilizzata sempre più spesso in diversi tipi di dispositivi.

La serie Cortex-M è composta da CPU con architettura ARMv6 M0, M0+ e M1, quelle con architettura ARMv7 M3, M4 e M7, e quelle con architettura ARMv8 M23, M33, M35, M35P, M52, M55 e M85. Proprio i microcontrollori ARM della serie Cortex-M, con architettura ARMv6 e ARMv7, è quella che oggi viene più spesso utilizzata a scopo hobbistico dagli appassionati di elettronica e dai makers.

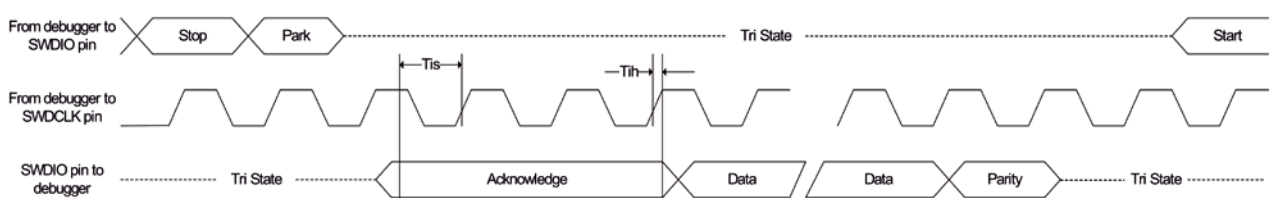
Già in passato la piattaforma Arduino aveva lanciato la scheda Arduino DUE con a bordo il microcontrollore Atmel ATSAM3X8E CPU ARM Cortex-M3 a 84 MHz e la scheda Arduino Zero prodotta da *arduino.cc*, identica ad Arduino M0 prodotta da *arduino.org*, con a bordo un microcontrollore Atmel SAMD21G18 CPU ARM Cortex-M0+ a 48 MHz. Per programmare i suddetti microcontrollori vergini, cioè privi di bootloader, nell'ambiente di sviluppo Arduino IDE abbiamo bisogno del programmatore Atmel-ICE debugger dal costo poco conveniente per un hobbista che deve caricare saltuariamente il firmware o il bootloader nel microcontrollore. Tutti i microcontrollori ARM per la programmazione e il debug utilizzano il protocollo JTAG (Joint Test Action Group) progettato principalmente per il test su chip e schede per valutare gli eventuali guasti in produzione. Necessita di 4 pin TCK, TDI, TMS e

Fig. 1
Andamento dei segnali sulla porta SWD durante la comunicazione in fase di lettura e scrittura.

Read Cycle



Write Cycle



TDO però l'ARM holding ha introdotto un secondo protocollo denominato SWD e può gestire le stesse funzionalità del protocollo JTAG con solo una coppia di piedini, il clock SWDCLK e un singolo pin di dati bidirezionale SWDIO, senza perdite di performance. È stato sviluppato con lo scopo di utilizzare il numero minore possibile di piedini per il debug, utile soprattutto per i microcontrollori con un numero limitato di pin.

SWDIO e SWCLK sono sovrapposti ai pin TMS e TCK, consentendo di utilizzare lo stesso connettore sia per il protocollo JTAG sia per il protocollo SWD. Nella **Tab. 1** viene riportata la descrizione di ogni singolo pin del connettore JTAG. Per comunicare con un dispositivo tramite SWD, i dati vengono inviati su SWDIO in modo bidirezionale sincrono al clock SWDCLK, come raffigurato nell'immagine in **Fig. 1**.

Come precedentemente detto per programmare i microcontrollori ARM della famiglia Atmel normalmente si usa il programmatore Atmel-ICE debugger, in alternativa però possiamo usare la scheda Arduino Zero Pro opportunamente modificata. Su di essa è presente il chip Atmel Embedded Debugger (EDBG) che fornisce un'interfaccia di programmazione e debug completa utilizzabile

PIN	TIPO	DESCRIZIONE
TCK	Input	Test clock
TDI	Input	Test Data Input
TMS	Input	Test mode select
TDO	Output	Test data output
nTRST	Input (opzionale)	Test reset

Tab. 1
Descrizione della piedinatura di un connettore JTAG.

direttamente nell'ambiente di sviluppo Arduino IDE senza la necessità di hardware aggiuntivo. La modifica è molto semplice e alla portata di tutti. Una volta in possesso della scheda Arduino Zero Pro dissaldare le due resistenze da 0 ohm, nello schema elettrico siglate con R104 e R105 e se assente, aggiungete il connettore SWD da 2 x 5 pin con passo 1,27 mm, come indicato in **Fig. 2**. In **Fig. 3** riportiamo la piedinatura del connettore SWD. Con questa semplice modifica e implementando sul PCB dei vostri prototipi un connettore SWD con passo 1,27 mm, potrete finalmente programmare in modo facilissimo e a basso costo i microcontrollori ATSAMD21G18 vergini. Con un cavetto a nastro (ribbon cable) a 10 poli, collegate

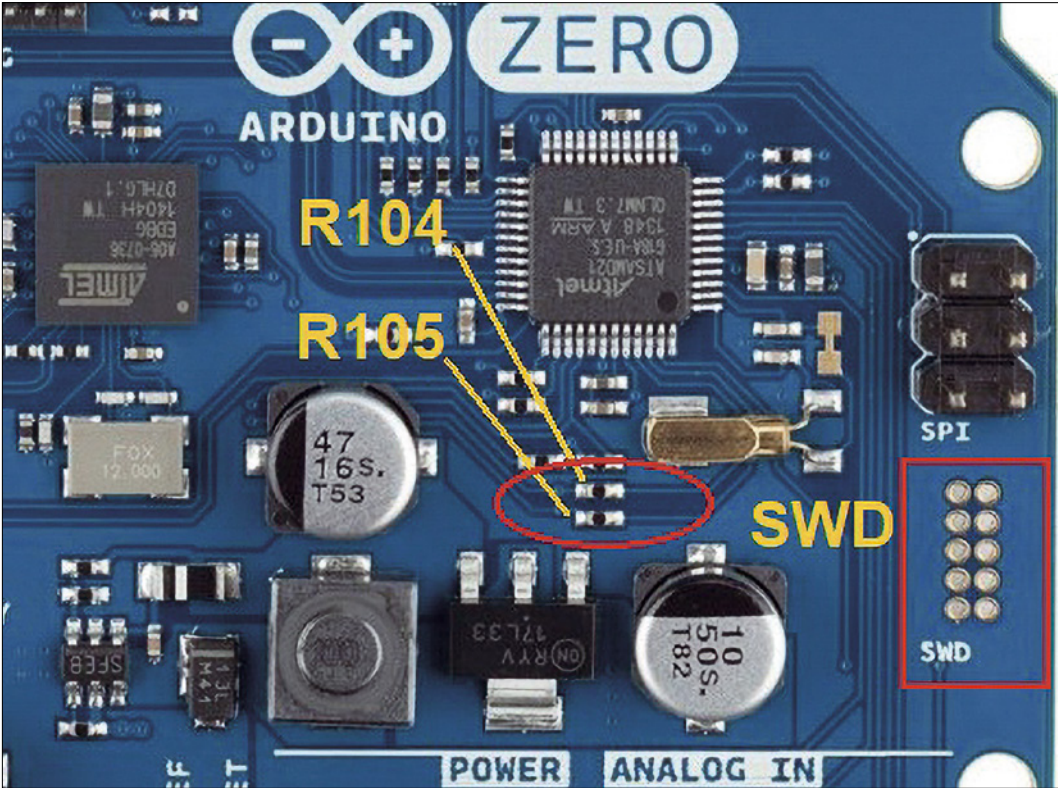
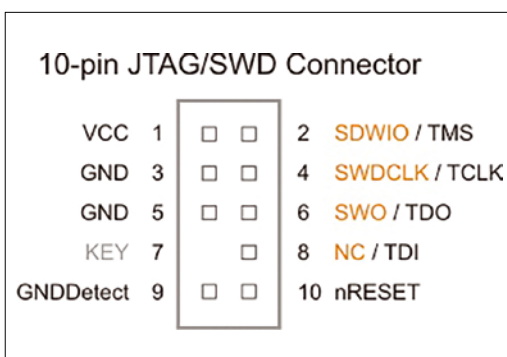


Fig. 2
Posizione delle due resistenze R104 e R105, e del connettore SWD sulla scheda Arduino M0/Zero Pro.

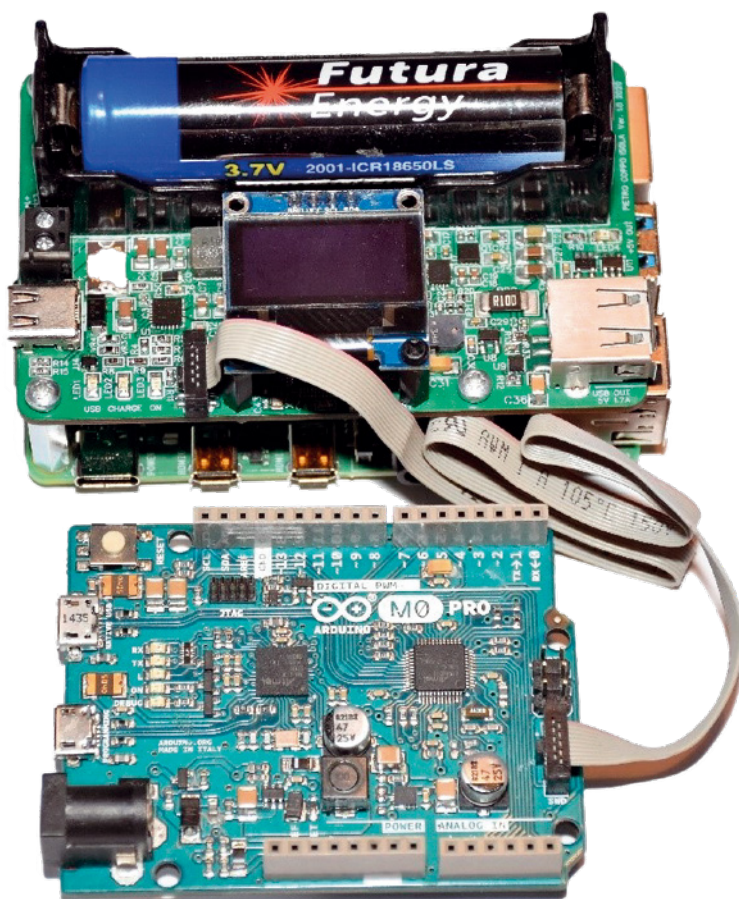
→ **Fig. 3**
Piedinatura del
connettore SWD
sulla scheda
Arduino.



la scheda da programmare alla scheda Arduino Zero Pro modificata come visibile in **Fig. 4** e mediante l'ambiente di sviluppo Arduino IDE la potrete programmare.

Il cavetto a 10 poli, utilizzato per la comunicazione tra le due schede, lo potete realizzare facilmente anche da soli utilizzando uno spezzone di cavo riciclando un vecchio cavo ribbon a 40 pin passo 1 mm, utilizzato per interfacciare i dischi rigidi IDE-ATA 2 con la scheda madre nei vecchi personal

↓ **Fig. 4**
Collegamento
mediante cavetto
SWD tra la scheda
Arduino Zero/
M0 modificata
e il prototipo da
programmare.



computer, di cui utilizzerete solo 10 conduttori. Alle due estremità basterà crimpare due connettori IDC da 10 pin passo 1,27 mm, facendo attenzione di posizionarli nel verso giusto cioè pin to pin. A differenza dell'ATSAMD21G18, l'ATSAM3X8E montato sulla scheda Arduino DUE possiede un bootloader pre programmato in fabbrica nella ROM. Quindi per caricare il firmware nella memoria Flash bisogna utilizzare un convertitore seriale USB TTL FT232, che supporti il funzionamento con i livelli logici a 3,3 V, collegando i piedini TX e RX sui piedini 27 (PA8 RX) e 2 (PA9 TX) del microcontrollore, impostando il baud rate a 1200 bps. Impostate ad alto livello il pin 130 PC0 "Erase", successivamente ponete a basso livello il pin 47 nRSTB "Reset" e dopo alcuni millisecondi lo impostate nuovamente a livello alto.

Dopo aver aspettato un secondo ponete il pin 130 "Erase" a basso livello. Per facilitare la procedura consigliamo di implementare sul PCB del prototipo dei pulsanti o dei jumper. A questo punto dalla console "CMD" di Windows, avviato con i privilegi di amministratore, utilizzando il programma open source *BOSSA* acronimo di "Basic Open Source SAM-BA Application" incluso nell'ambiente di sviluppo Arduino IDE versione 1.6.1 oppure scaricabile dalla pagina di presentazione di questo numero, lanciate il comando:

```
bossac.exe -i -d --port=COM3 -U true -e -w -v -R -b sketch.bin
```

dove COM3 corrisponde al numero della porta COM assegnata dal sistema operativo al vostro convertitore USB TTL. Facciamo presente che il programma *bossac.exe* e il sorgente *sketch.bin* devono trovarsi nella stessa cartella.

Per facilitare la programmazione, sulla scheda Arduino DUE hanno aggirato questo problema inserendo, in ogni sketch caricato sul micro, un codice che gira in background il quale controlla l'apertura e la chiusura della porta nativa con un baud rate a 1200 bps, se ciò avviene avvia un "soft erase" cancellando la Flash e successivamente lancia il bootloader.

Nel caso la programmazione non è terminata correttamente o il sorgente caricato nella Flash sia corrotto, il microcontrollore resta bloccato e la programmazione mediante la porta USB nativa non funzionerà più. Per tale motivo hanno inserito sulla scheda un micro ATMEGA16u2 che fa capo alla seconda porta USB "programming", il quale controlla direttamente i piedini Erase e nRSTB



Fig. 5
Piedinatura del connettore a 20 pin presente sul programmatore ST-LINK/V2 e sul J-link.

dell'ATSAM3X8E ponendoli correttamente ad alto o basso livello, permettendo così la programmazione mediante l'ambiente di sviluppo Arduino IDE tramite i piedini TTL RX e TX.

Purtroppo questi due microcontrollori nei progetti odierni vengono usati sempre meno poiché surclassati dai microcontrollori STM32 della ST Microelectronics molto più veloci e performanti. L'STM32 è la serie di microcontrollori a 32 bit basati sull'architettura ARM Cortex-M che ha ottenuto il maggior successo nell'industria, ed è stato stimato che circa la metà di tutti dispositivi industriali venduti si basa sulla famiglia STM32. Questo è il motivo che ci ha spinto a scrivere questo articolo per facilitare il neofiti nella programmazione di questi microcontrollori nell'ambiente di sviluppo Arduino IDE.

Incominciamo con l'analizzare il primo programmatore, quello prodotto dalla ST Microelectronics ST-LINK/V2 un debugger e programmatore in circuit che possiede due connettori, uno a 4 pin per la programmazione dei microcontrollori a 8 bit STM8 e un connettore JTAG/SWD a 20 pin per la programmazione dei microcontrollori a 32 bit STM32, la cui piedinatura è riportata in **Fig. 5**. Per la programmazione dei microcontrollori STM32 abbiamo bisogno di soli cinque dei 20 pin del connettore. Il pin 7 SWDIO (serial wire data input output), il pin 9 SWCLK (serial wire clock), il pin 15 nRST (reset attivo a GND), il pin 20 GND e il pin 1 VAPP. Il piedino 13 SWO (serial wire output) non serve per la programmazione poiché usato unicamente come *trace port* in fase di *debug*.

A scopo informativo, per chi non lo sapesse, il piedino VAPP non fornisce nessuna alimentazione in uscita, cioè non può essere usato per alimentare il circuito da programmare. Viene collegato al polo positivo di alimentazione del prototipo da programmare affinché il programmatore ST-Link/V2 possa

rilevare il voltaggio e impostare la tensione appropriata sulle uscite durante la programmazione. Poiché l'acquisto del programmatore ST-LINK/V2, per un neofita risulterà poco conveniente come alternativa più economica ci sentiamo di suggerire il programmatore Nucleo ST-Link, presente su tutte le schede Nucleo-64 e Nucleo-144, come quelle presenti a catalogo futura shop Nucleo-L152RE codice 7309-NUCLEOL152RE e Nucleo-F401RE codice 7309-NUCLEOF401RE.

Come spiegato nel manuale del costruttore per utilizzarle come programmatore **è necessario rimuovere i ponticelli del connettore siglato CN2, sulle schede Nucleo-64, oppure siglato CN4, sulle schede Nucleo-144.**

Una volta tolti i ponticelli, per programmare un microcontrollore STM32, utilizzate il connettore DuPont a 6 pin siglato CN4, sulle schede Nucleo-64, oppure quello siglato CN6, sulle schede Nucleo-144. La piedinatura di questo connettore è riportata nella **Fig. 6**.

Dal 2021 Arduino e l'azienda Segger hanno annunciato una partnership che finalmente permette agli sviluppatori di utilizzare il programmatore universale e strumento di debug J-link anche nell'ambiente di sviluppo Arduino IDE rendendo così più facile il debug delle schede Arduino di ultima generazione come la scheda *Portenta*. Per un uso non commerciale, cioè prettamente hobbistico, l'azienda Segger mette a disposizio-

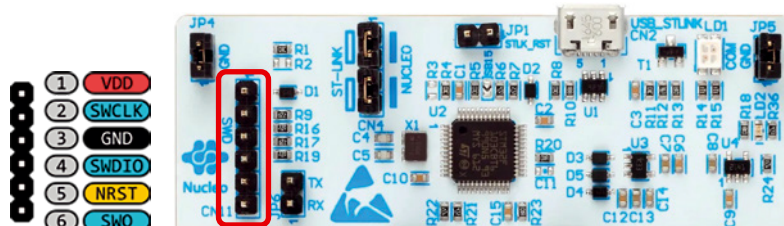
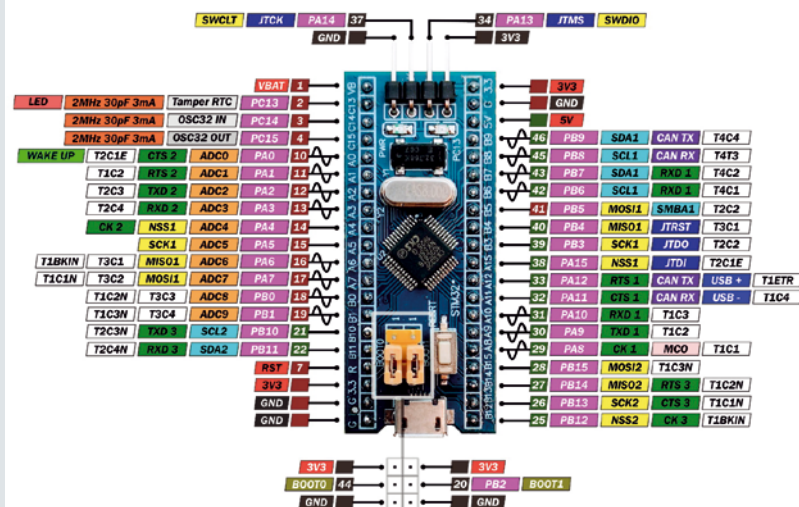


Fig. 6
Piedinatura del connettore SWD nelle schede Nucleo ST-Link.

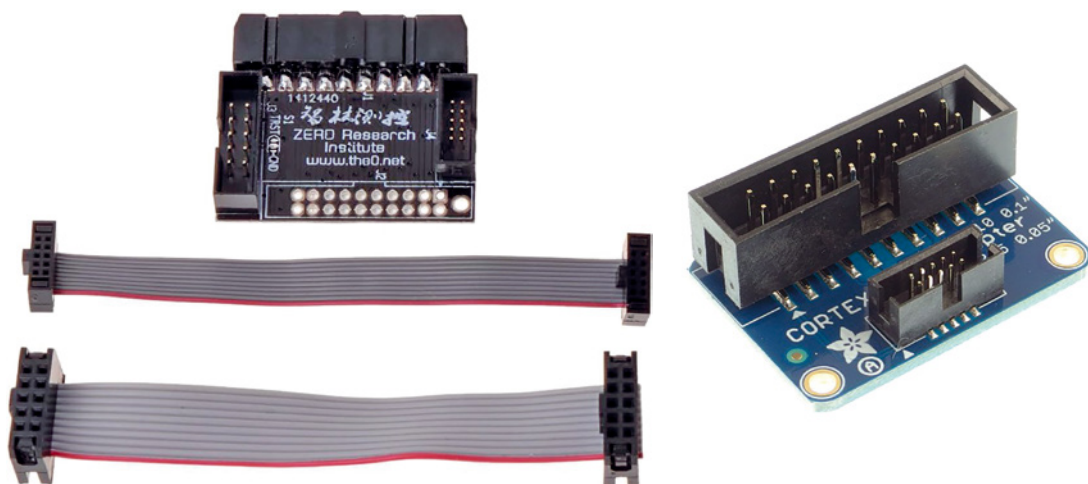


PIN	NOME	DESCRIZIONE
1	CPU	STM32F103C8T6
2	Memoria Flash	64/128KB
3	Memoria SRAM	20KB
4	Frequenza di clock	72MHz(Max)
5	Entrate analogiche	14
6	Pin digitali I/O	37
7	Pin PWM	12
8	Periferiche I ² C	2
9	Periferiche SPI	2
10	Periferiche CAN 2.0	1
11	Alimentazione	2.7V to 3.6V
12	Massima corrente sui pin	6mA

ne per gli studenti e i makers il programmatore J-link EDU a un prezzo veramente economico se confrontato con quello della versione commerciale J-link PRO. J-link è lo strumento di programmazione e debug più utilizzato sul mercato, che possiede funzionalità uniche permettendo l'utilizzo di tutti i principali strumenti di sviluppo in commercio. Momentaneamente l'ambiente di sviluppo Arduino IDE permette l'utilizzo del J-Link solo con le schede Arduino provviste di microcontrollori ARM, cioè non quelle basate su microcontrollori AVR a 8 bit, e le schede provviste di microcontrollori Nordic NRF52. È un prodotto molto valido e versatile consigliato per i più esperti che cercano un programmatore per microcontrollori universale.

Possiede un connettore a 20 pin JTAG/SWD la cui piedinatura è riportata in **Fig. 5**. Considerato che per la programmazione dei microcontrollori ARM di solito viene usato il protocollo SWD con connettori a 10 pin, come quello in **Fig. 3**, vi informiamo che in commercio potete trovare degli adattatori che vi permettono di convertire il connettore JTAG a 20 pin in un connettore SWD a 10 pin, come quello visibili in **Fig.7**. Sapendo che ad un hobbysta o a un maker piace smanettare e riciclare vecchie schede elettroniche, per la programmazione dei microcontrollori STM32 vogliamo proporvi un'alternativa ancora più economica, basata sulla trasformazione di una scheda STM32 Blue Pill in un programmatore ST-Link/V2.

Fig. 7
Adattatori JTAG 20
pin a SWD 10 pin.



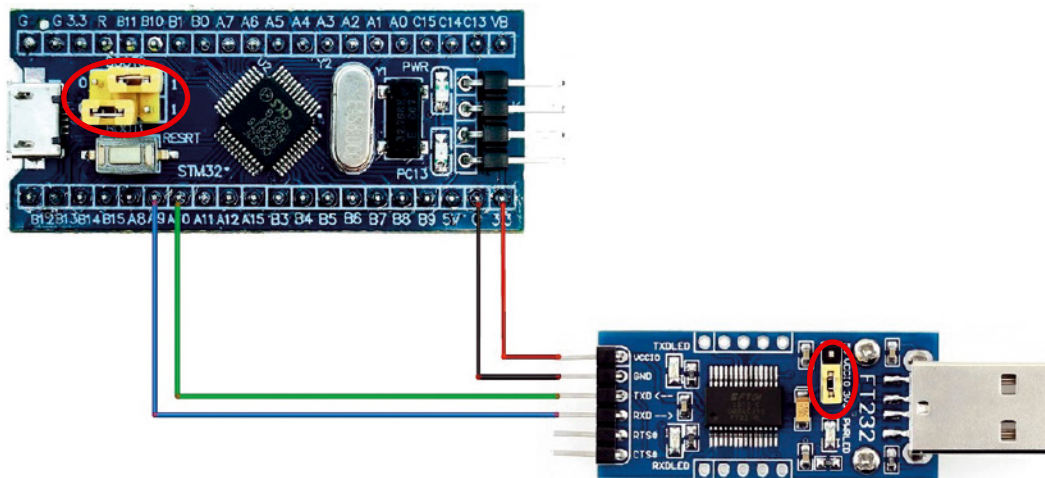


Fig. 8
Cablaggio
per il colle-
gamento del
convertitore
USB TTL alla
Blue Pill.

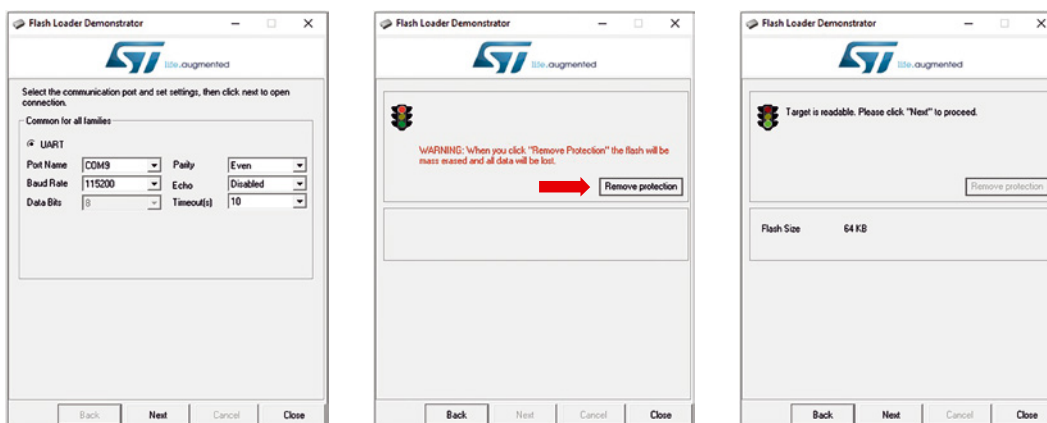


Fig. 9
Configurazio-
ne della porta
seriale nel
programma
*Flash loader
demostrator*.

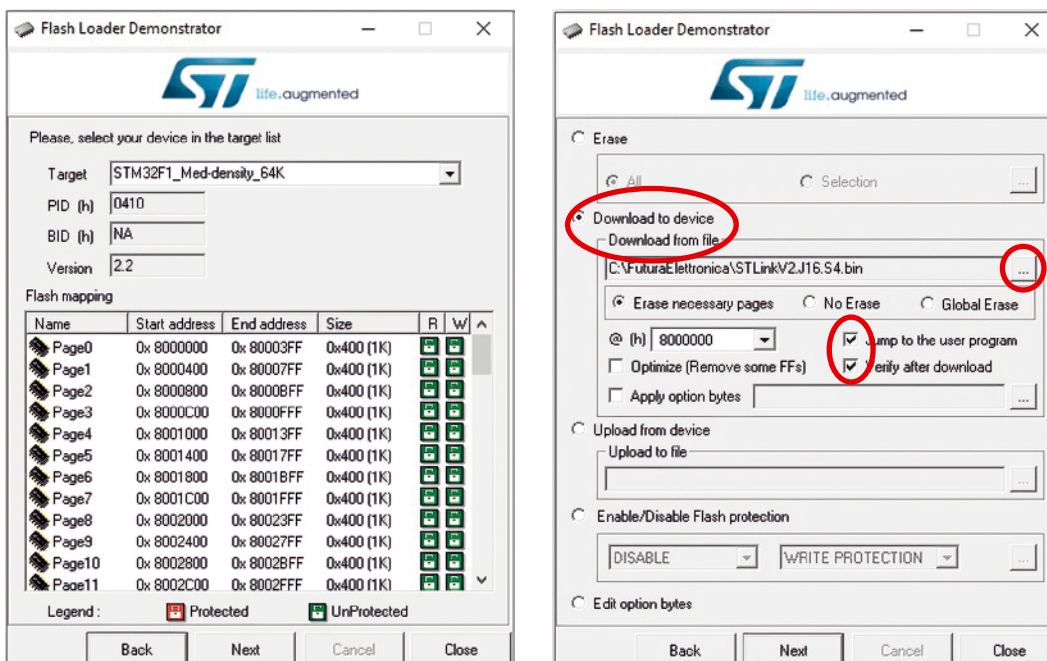


Fig. 10
Impostazioni
di program-
mazione
della scheda
Blue Pill nel
programma
*Flash loader
Demostrator*.

Fig. 11
Fasi del
caricamento del
sorgente nella Flash
della Blue Pill.

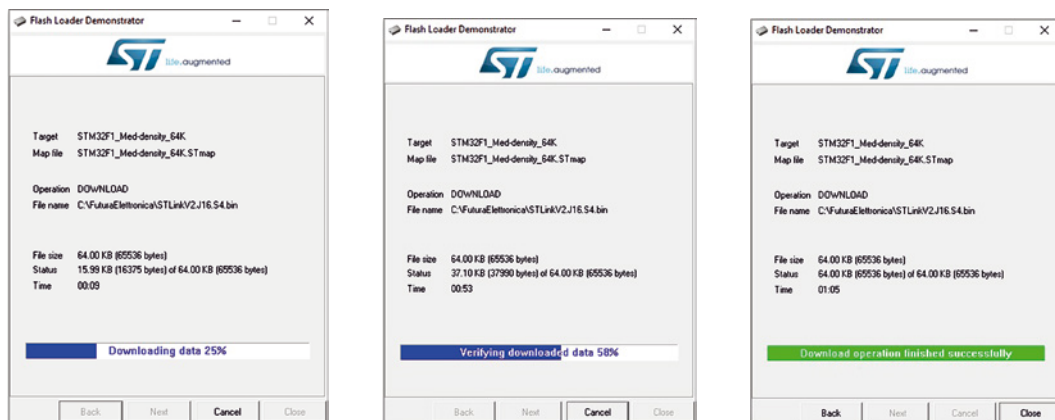


Fig. 12
Aggiornamento del
firmware della Blue
Pill modificata in
ST-Link.

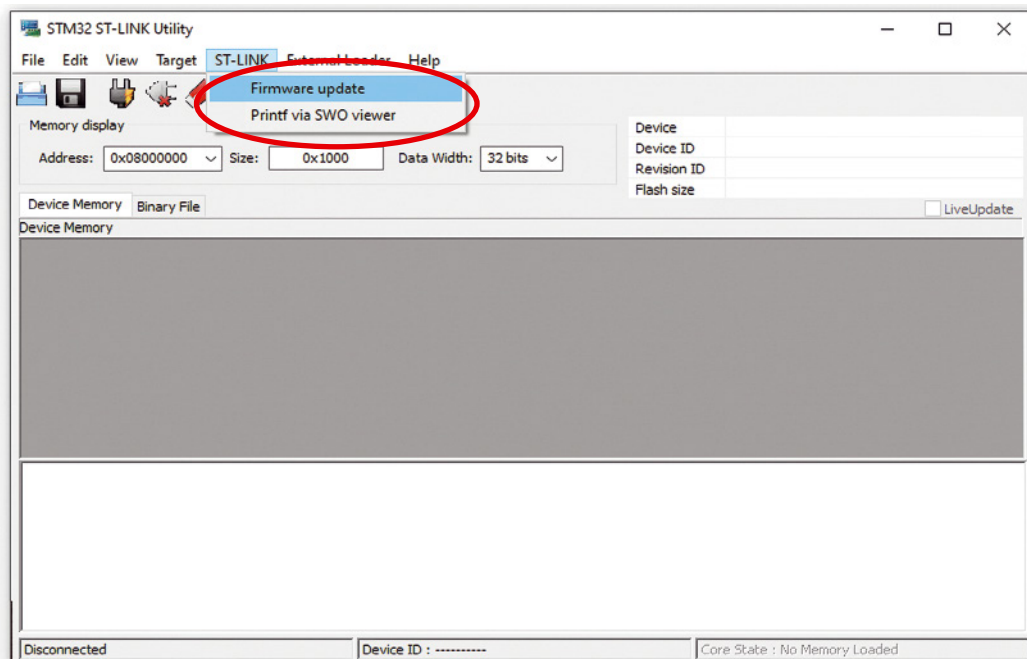
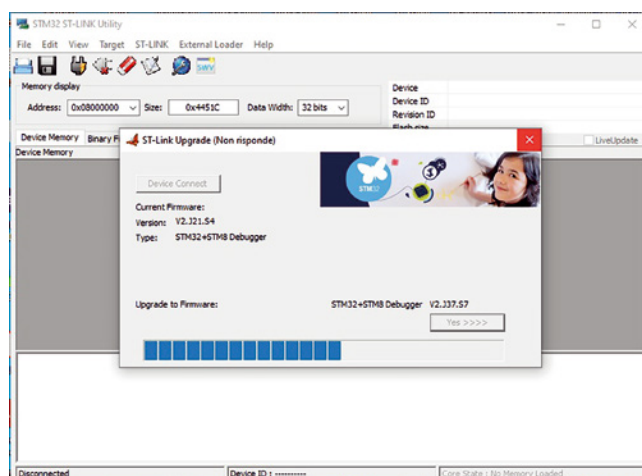
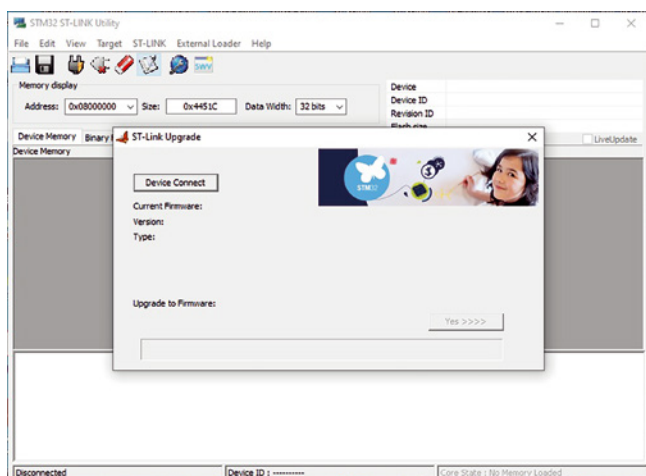


Fig. 13
Finestra per
l'aggiorna-
mento del
firmware
della Blue Pill
modificata in
ST-Link.



MODIFICA DELLA BLUE PILL

Procuratevi una scheda STM32 Blue Pill a catalogo Futura Elettronica con il codice 2846-STM32ARMCORE, un modulo convertitore seriale USB TTL FT232 compatibile con i livelli logici a 3,3 V (codice 2850-FT232USBUARTA), il programma *Flash_loader_demo_v2.8.0.exe* e il programma *STM32 ST-Link Utility*, entrambi scaricabili dal nostro sito www.elettronica.in.it. Installate e lanciate sul vostro PC il programma *Flash_loader_demo_v2.8.0.exe* ricordando di impostare correttamente sulla scheda Blue Pill il jumper BOOT 0 sulla posizione 1 e sul convertitore USB TTL il jumper della selezione del voltaggio su 3,3 V, e collegate i rimanenti piedini seguendo il cablaggio riportato in **Fig. 8**.

Dal menu a tendina del programma *Flash loader demonstrator*, lanciato dal menu *Start* di Windows con il nome *Demonstrator GUI*, selezionate la porta COM assegnata dal sistema operativo al convertitore USB TTL, impostate correttamente i parametri di comunicazione come visibile nella prima immagine in **Fig. 9**, e schiacciate il pulsante *Next*. Verrà visualizzata la finestra visibile nella terza immagine della **Fig. 9**. In caso contrario ricontrollate il corretto cablaggio. Gli errori più comuni sono l'inversione dei collegamenti tra i piedini di comunicazione RX e TX del convertitore USB TTL FT232 e i piedini A9 TX e A10 RX della Blue Pill oppure lo scorretto posizionamento del jumper BOOT 0. Se in passato avete già programmato la Blue Pill, utilizzandola in qualche progetto, probabilmente è attivata la protezione di scrittura della memoria Flash. In questo caso apparirà una schermata simile a quella visibile nella seconda immagine della **Fig. 9**.

Basterà cliccare sul pulsante *Remove protection*, come indicato dalla freccia rossa, e apparirà la schermata visibile nella terza immagine. Se questo non funziona in alternativa potete usare il programma *STM32 ST-Link Utility* oppure la nuova versione *STM32CubeProgrammer*. Cliccando sul pulsante *Next* si aprirà la finestra visibile nella prima immagine di **Fig. 10**, controllate che nel menu a tendina *Target* sia selezionato il dispositivo corretto. Cliccando sul pulsante *Next* si aprirà la finestra visibile nella seconda immagine. Selezionate il pulsante di azione *Download to device* e nella casella di testo *Download from file* selezionate il percorso del firmware *STLinkV2.J16.S4.bin* da caricare nella Flash della Blue Pill. Selezionate le caselle *Jump to the user program* e *Verify after download* come visibile nelle seconda immagine di **Fig. 10**. Cliccando sul pulsante *Next* apparirà la schermata

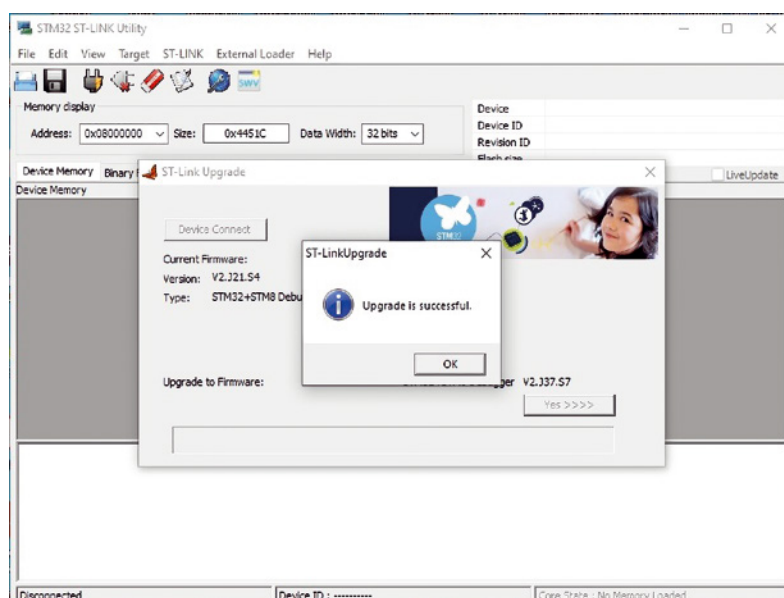


Fig. 14
Schermata di conferma del corretto aggiornamento del firmware.

di cancellazione della memoria Flash e successivamente quella di caricamento visibile nella prima immagine di **Fig. 11**. Una volta caricato il file binario apparirà una nuova schermata per la verifica della corretta programmazione, come quella visibile nella seconda immagine.

Se la programmazione è andata a buon fine apparirà la schermata visibile nell'ultima immagine di **Fig. 11**. Cliccate sul pulsante *Close* per chiudere il programma. A questo punto dovete aggiornare il firmware dell'ST-LINK.

Installate il programma *STM32 ST-Link Utility*

Fig. 15
Schermata di conferma del corretto aggiornamento del firmware.

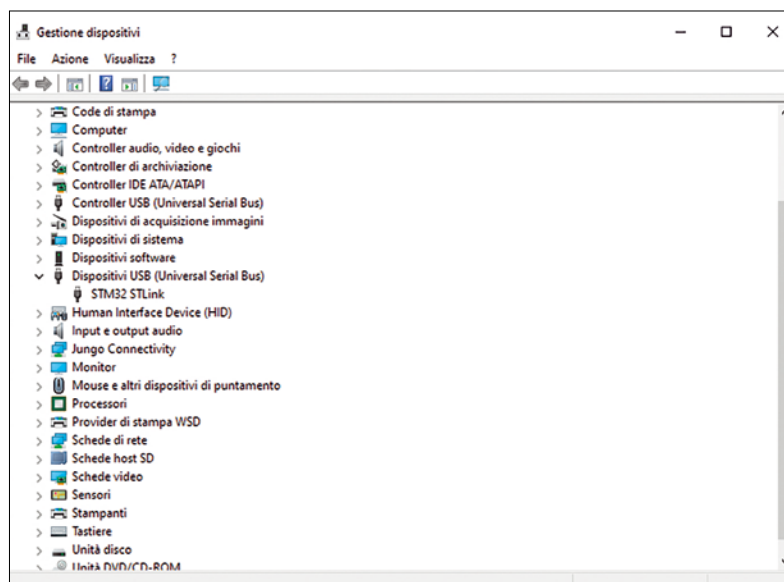
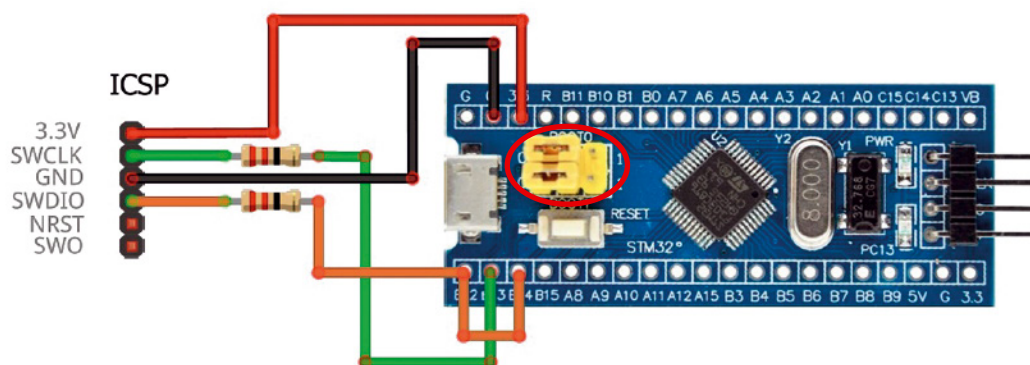


Fig. 16
Cablaggio del
programmatore
Blue Pill ST-Link/
V2 al connettore
ICSP del PCB
del progetto da
programmare.
Posizionare il
jumper BOOT 0
sulla posizione 0.



oppure quello più recente che lo sostituisce
STM32CubeProgrammer.

Durante l'installazione sul sistema operativo
Windows vi verrà chiesto di installare i driver per il
programmatore ST-LINK. Se questo non avviene
significa che i driver sono già installati, oppure
a causa di un errore sarete costretti a installarli
manualmente.

Anche se la procedura di aggiornamento firmwa-
re è simile a quella utilizzando il programma
STM32CubeProgrammer, in questo articolo spiegheremo
come utilizzare il programma *STM32 ST-Link
Utility* il cui utilizzo è molto più semplice. Una volta
lanciato il programma, cliccate sul menu *Target*,
selezionate *Connect* e successivamente nel menu
ST-LINK selezionate *Firmware update*, come visibile
in **Fig. 12**. Verrà visualizzata una nuova schermata
come quella visibile nella prima immagine di **Fig.**
13. Cliccate sul pulsante *Device Connect* e si aprirà
una schermata come quella visibile nella seconda
immagine della **Fig. 13**. Cliccando sul pulsante "Yes
>>>" verrà visualizzata la schermata di **Fig. 14**,
confermate cliccando sul pulsante *OK* e chiudete il
programma.

Se lanciate *Gestione dispositivi* dal *Pannello di
controllo* di Windows, nella sezione *Dispositivi USB*
dovrebbe apparire il dispositivo *STM32 ST-Link*,
come visibile in **Fig. 15**.

Terminata la programmazione della Blue Pill non
scordate di posizionare il jumper BOOT 0 sulla
posizione 0, cortocircuitate i piedini B12 e B14 e
collegate la scheda al connettore ICSP presente
sul PCB del vostro prototipo da programmare se-
guendo il cablaggio di **Fig. 16**. Le due resistenze da
22 ohm in serie alle linee SWCLK e SWDIO servono
come protezione e come limitatori di rumore du-
rante la comunicazione ad alta velocità limitando le
riflessioni del segnale.

È sempre consigliato che le linee JTAG e SWD
abbiano queste resistenze in serie anche se molti
progettisti le trascurano.

Terminato il cablaggio nel sottomenu *Upload
Method* del menu *Strumenti* dell'ambiente di
sviluppo Arduino IDE scegliete il programmatore
STM32CubeProgrammer(SWD) e lanciate il comando
per caricare il firmware.

CONCLUSIONI

Sperando che con questo articolo abbiamo sod-
disfatto tutti i lettori che da tempo desideravano
cimentarsi nella programmazione dei microcon-
trollori basati sull'architettura ARM Cortex-M,
senza per questo essere costretti ad acquistare
un ulteriore programmatore che tra l'altro sarà
utilizzato un paio di volte all'anno.
Inoltre riciclando le vecchie schede che restano
inutilizzate in qualche cassetto o scaffale del
laboratorio, come nel nostro caso la scheda Blue
Pill, non solo risulta economicamente convenien-
te ma allo stesso tempo anche ecologicamente
conveniente poiché in questo modo contribuiamo
a ridurre l'impatto ambientale. Detto questo, non
ci resta che congedarci da voi e augurarvi una
buona programmazione. □



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
NUCLEOL152RE	Nucleo L152RE kit di sviluppo micro STM	19,90
NUCLEOF401RE	Nucleo F401RE kit di sviluppo micro STM	27,90
STM32ARMCORE	Board con Microcontrollore STM32F103C8T6	14,80
FT232USBUART	Convertitore USB-UART	12,80

IL MATERIALE VA RICHIESTO A:

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

INDISPENSABILI IN LABORATORIO!



Tappeto in silicone per saldatura 550x350 mm

PER SALDARE

Tappeto in silicone antistatico multifunzione per saldatura resistente al calore fino a una temperatura di 500°C. Estremamente flessibile, anche dopo piegato ritorna sempre alla sua forma originale. Dispone di 3 piccoli contenitori e di 5 aree magnetiche.



cod. AS12

€ 35,00



Tappetino antistatico 550x300x2 mm

Tappetino antistatico particolarmente indicato per posizionare e manipolare componenti o schede elettroniche. Viene fornito completo di cavo di collegamento messa terra. Resistività superficiale: $10^7 \sim 10^{10}$ ohm/m², tempo decadimento carica statica: <0,03 secondi.

cod. AS13

€ 13,50

ANTISTATICO



Tappetino da taglio 300x220x2,8 mm

PER TAGLIARE

Tappetino da taglio realizzato in PVC (Polivinil-cloruro) ideale come piano di lavoro per tagliare carta, cartoncino, cartone pressato, fogli di vinile, ecc. Dispone di griglia di misurazione 1x1 cm e goniometro che consente all'utente di eseguire segni e tagli precisi.



cod. CUTMAT30X22

€ 9,00



Scopri la gamma completa su www.futurashop.it

NUOVI ALIMENTATORI SWITCHING DA RETE

COD. ER24W24V

€ 10,⁵⁰



ALIMENTATORE SWITCHING
24V / 1A - 24W

Tensione di ingresso: da 90 a 240 VAC - 50/60 Hz
Potenza: 24 W
Tensione in uscita: 24 Vdc
Massima corrente di uscita: 1 A
Tensione di ripple: 200 mVpp
Consumo di potenza a vuoto (max): 0.075 W
Lunghezza del cavo: 1,5 metri
Plug: 5,5 x 2,1 mm, positivo centrale
Dimensioni (mm): 82,3 x 32,6 x 74,1 (L x W x H)
Peso: 112 grammi

ALIMENTATORE SWITCHING
6V / 1A - 6W

Tensione di ingresso: da 90 a 240 VAC - 50/60 Hz
Efficienza: 75%
Potenza: 6 W
Tensione in uscita: 6 Vdc
Massima corrente di uscita: 1 A
Tensione di ripple: 200 mVpp
Consumo di potenza a vuoto (max): 0.075 W
Lunghezza del cavo: 1,5 metri
Plug: 5,5 x 2,1 mm, positivo centrale
Temperatura di lavoro: 0°C ÷ 40°C
Dimensioni (mm): 38 x 25 x 69,6 (L x W x H)
Peso: 80 grammi

COD. ER06W06V

€ 7,⁰⁰

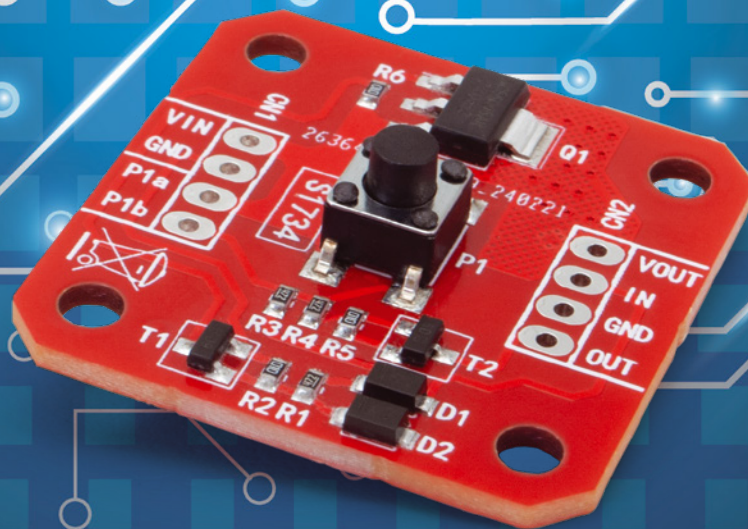


Prezzi IVA inclusa.

**VASTA GAMMA DI MODELLI
PER TUTTE LE ESIGENZE**

DISPONIBILI ANCHE CON LE SEGUENTI CARATTERISTICHE:

CODICE		TENSIONE D'INGRESSO	TENSIONE D'USCITA	CORRENTE D'USCITA	POTENZA	RIPPLE	DIMENSIONI	PREZZO
ER09W09V		90 a 240 VAC	9 VDC	1 A	9 W	150 mVpp	76,2 x 27,6 x 61,6 mm	7,00
ER12W12V		90 a 240 VAC	12 VDC	1 A	12 W	150 mVpp	76,2 x 27,6 x 61,6 mm	7,50
ER15W15V		90 a 240 VAC	15 VDC	1 A	15 W	200 mVpp	79,2 x 27,6 x 69,1 mm	8,50
ER24W12V		90 a 240 VAC	12 VDC	2 A	24 W	120 mVpp	82,3 x 32,6 x 74,1 mm	11,50



LATCH DI ALIMENTAZIONE

di BORIS LANDONI

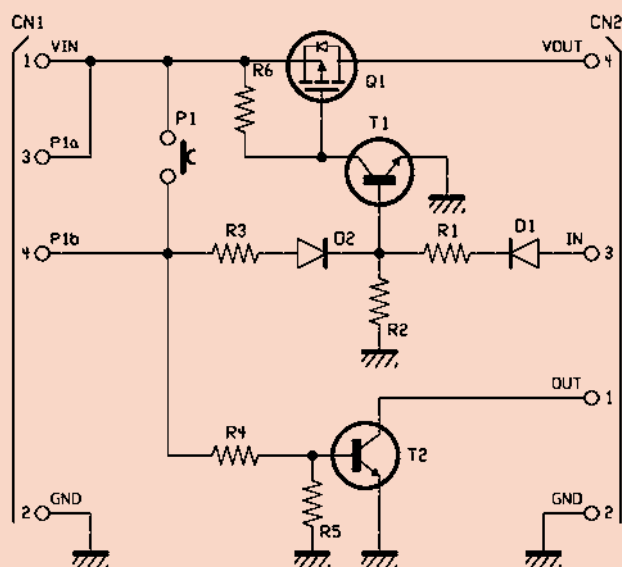
Permette di alimentare circuiti elettronici con la semplice pressione di un pulsante, intervenendo direttamente sull'alimentazione principale e consumando nulla quando è a riposo.

P

er fornire alimentazione a un circuito elettronico c'è solamente una soluzione: utilizzare un interruttore elettromeccanico oppure un relé azionato da un interruttore a stato stabile; tutti i dispositivi che si accendono con la pressione di un pulsante presuppongono, invece, l'esistenza di una sezione

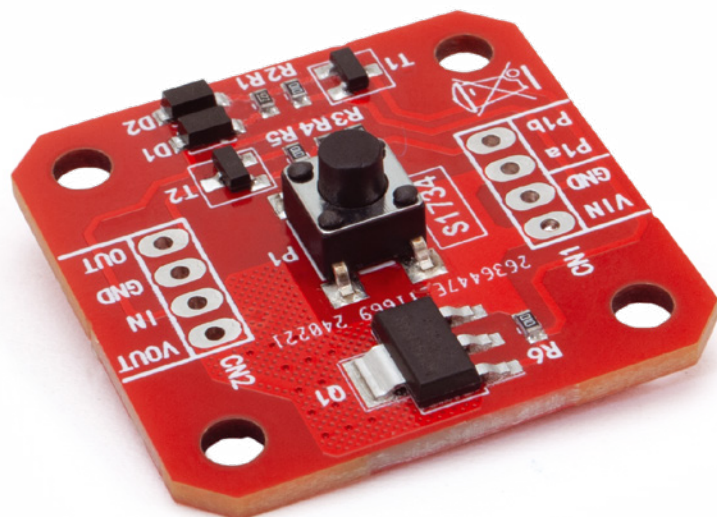
circuitale costantemente alimentata che al comando del pulsante stesso interviene su un interruttore statico oppure su un relé, per fornire l'alimentazione alla parte circuitale che normalmente deve rimanere inattiva per ragioni di contenimento dei consumi.

Funzionano così tutti i dispositivi e gli apparati che ci circondano, dai Personal Computer con alimentazione ATX ai notebook, ai tablet e smartphone, che anche quando li credia-



mo spenti in realtà non lo sono e possono, se n'è discusso a lungo tra esperti di sicurezza, mantenere attivi sistemi in grado di trasmettere in remoto riprese audio e della fotocamera. Per i dispositivi alimentati dalla rete elettrica che si accendono con un pulsante, proprio per limitare il consumo dovuto alla parte circuitale sempre alimentata sono state varate normative europee che dal 2013 impongono che il consumo in modalità standby non superi la soglia di 0,5 watt, che salgono a 1 W se gli apparecchi standby consentono di visualizzare il proprio stato o informazioni specifiche. Nel 2023 sono state varate nuove norme UE che stabiliscono nuove limitazioni, cioè

↓
Il prototipo del circuito latch montato.



dal 2025 i dispositivi non dovranno consumare più di 0,5 Watt in standby o in modalità spenta, oppure 0,8 Watt se sono in standby mentre visualizzano informazioni, poi dal 2027 i dispositivi non dovranno consumare più di 0,5 Watt in standby, 0,3 Watt in modalità spenta o 0,8 W se sono in standby mentre visualizzano il proprio stato.

È tuttavia possibile sviluppare particolari circuiti che consentono di avere un assorbimento nullo a riposo adottando particolari soluzioni circuitali basate su un circuito di attivazione funzionante a latch, ossia capace di innescare un dispositivo che chiude fisicamente l'alimentazione e che una volta alimentato si autosostiene, potendo poi, con una logica o un'azione via software, disconnettersi in maniera automatica.

La soluzione proposta in queste pagine è qualcosa di simile, tuttavia parte dalla presenza di un'alimentazione di standby e funziona da latch, ossia fornisce al circuito che alimenta una tensione che gli permette poi di mantenerlo attivo. Possiamo quindi considerare il circuito come una sorta di relé ad autoritenuta, tuttavia non tradizionale ma allo stato solido, adatto a gestire un ampio campo di tensioni continue che va da 5 a 30 volt e idoneo ad alimentare dispositivi e circuiti elettronici (anche e soprattutto logici) che una volta alimentati sono in grado di mantenersi accesi per il tempo che occorre loro.

SCHEMA ELETTRICO

Andiamo dunque a vedere di cosa si tratta analizzandone lo schema elettrico, da cui appare la grande semplicità dell'insieme, che si riduce a un MOSFET, due transistor bipolari NPN, altrettanti diodi al silicio e qualche resistenza.

Il circuito latch (così chiamato perché una volta innescato rimane tale) va abbinato a un dispositivo elettronico, a una rete logica (che può essere anche solo semplicemente un multivibratore bistabile composto da porte NAND...) o anche a una scheda Arduino e richiede un'alimentazione di standby in corrente continua, che può essere ricavata da un comune alimentatore switching di piccolissima potenza: anche meno di 1 watt.

Vediamo come funziona premettendo che il connettore CN1 corrisponde al lato di entrata e che CN2 è quello di uscita della tensione di accensione (pin 4, VOUT) e di ingresso del segnale di autoritenuta (pin 3, IN). Dunque, a CN1 si collega l'alimentazione in standby, vale a dire quella sempre presente, mentre a CN2 si connette il dispositivo da alimentare, che poi provvederà da sé a mantenere

Listato 1

```
#define PUSH_PRESSED 2
#define MAINTAIN_ON 3
#define LED 13

void setup()
{
  pinMode(PUSH_PRESSED, INPUT_PULLUP);
  pinMode(MAINTAIN_ON, OUTPUT);
  pinMode(LED, OUTPUT);
  //Se sto avviando, significa che il pulsante è stato premuto, quindi mantengo l'alimentazione attiva
  digitalWrite(MAINTAIN_ON, HIGH);
  digitalWrite(LED, HIGH);
  while (digitalRead(PUSH_PRESSED) == LOW) delay(10);
}

void loop()
{
  if (digitalRead(PUSH_PRESSED) == LOW)
  {
    digitalWrite(LED, LOW);
    digitalWrite(MAINTAIN_ON, LOW);
    while (digitalRead(PUSH_PRESSED) == LOW) delay(10);
  }
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}
```

innescato il MOSFET per avere costantemente alimentazione fino al cambiamento delle condizioni di funzionamento.

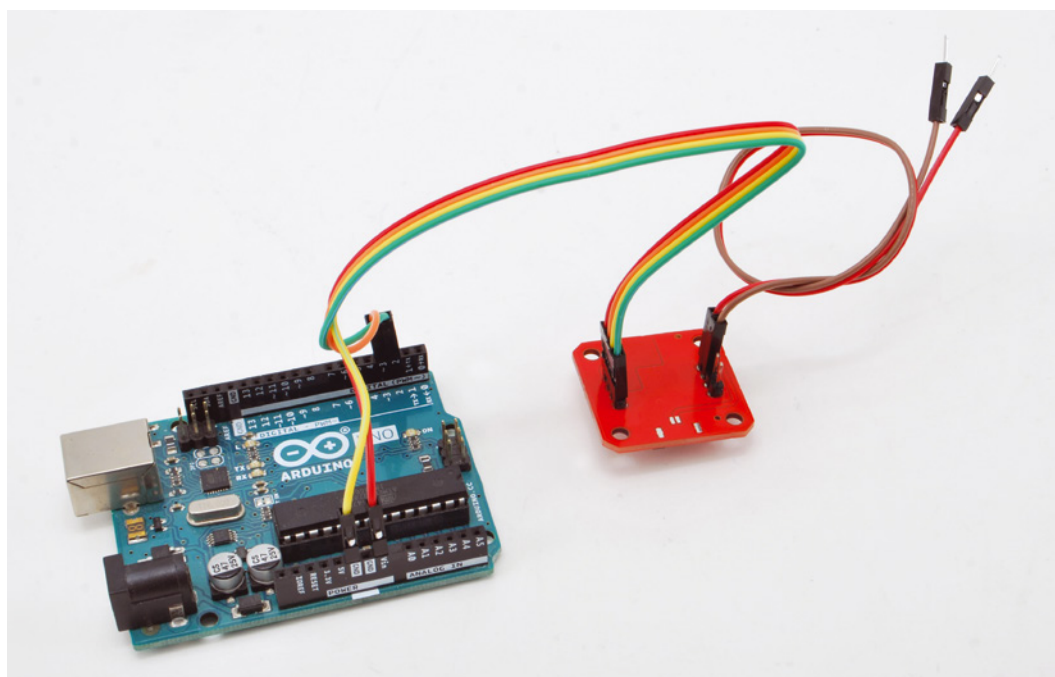
L'alimentazione "sempre presente" si applica tra i poli 1-3 e 2 del connettore CN2, rispettivamente positivo e massa, mentre il contatto 4 è stato pensato per il controllo in tensione del circuito, alternativo al pulsante P1.

Dunque, alimentando CN1 il circuito resta a riposo fin quando non si preme il pulsante P1, ovvero si fornisce una tensione tra il contatto 4 e GND che può essere il livello logico reso disponibile dall'uscita di un microcontrollore o di una porta logica; quando ciò avviene, il transistor NPN T1, collegato come interruttore statico (a emettitore comune) viene polarizzato dalla corrente che attraverso la resistenza R3 e il diodo D2 fluisce nella sua base, per effetto della connessione a partitore con R2. Il collettore del T1 si porta a un potenziale di pochissime centinaia di millivolt, il che determina la polarizzazione fino allo stato di ON del MOSFET a canale P siglato Q1, fino all'istante prima interdetto (si tratta di un MOSFET enhancement-mode...) perché il suo gate era mantenuto equipotenziale rispetto al source per effetto della resistenza R6. Ora il drain del Q1 si trova allo stesso potenziale dei contatti 1 e 3 del CN1 e la porta al 4 del CN2, da dove può alimentare il circuito da "accendere".

Notate che la pressione del pulsante fa andare in saturazione anche un secondo transistor, il T2, per effetto della polarizzazione operata dal partitore resistivo R4-R5; il collettore di detto NPN porta a livello basso il contatto OUT del CN2 e può essere utilizzato per triggerare una logica bistabile, ovvero fornire a un microcontrollore, il cui pin d'ingresso sia provvisto di pull-up, la condizione di richiesta di accensione, ovvero per intervenire su una rete CR di power-on-reset. La stessa uscita è operativa sempre, a prescindere dall'attività della linea di alimentazione Vout, vale a dire che se quando il MOSFET è in conduzione e Vout porta all'utilizzatore la tensione, ogni pressione ulteriore del pulsante o applicazione di un livello logico alto al piedino 4 del connettore CN1, determina la saturazione del transistor T2 e il livello logico; questo consente di implementare funzioni come il reset del contatore che lascerebbe spegnere il circuito, ovvero dei timer retriggerabili per prolungare il tempo di funzionamento del circuito che il latch accende e mantiene alimentato.

Per operare l'autoritenuta, il circuito che il nostro latch alimenta deve fornire al contatto IN un livello logico alto (da 5 a 12V) in modo da mantenere polarizzata la base del T1 anche a pulsante rilasciato; questo può essere ottenuto facilmente con una linea di uscita di un microcontrollore o con una

➔ **Fig. 1**
Connessione del
circuit latch ad
Arduino UNO per
eseguire lo sketch
di test.



logica bistabile, resettabile da altra logica cablata o da un microcontrollore, ovvero dal verificarsi di un evento nel circuito che viene alimentato, come ad esempio il consenso al termine di una sequenza di operazioni che il circuito deve effettuare una volta alimentato. Ma questo sta un po' all'uso che farete del nostro latch.

Al rilascio del pulsante, la tensione presente sul piedino IN del connettore CN2 mantiene in saturazione il transistor T1 e quindi in stato di ON il MOSFET; se tale tensione viene meno, il MOSFET si interdice e il circuito a valle del latch si spegne. A proposito di tale sezione del circuito, notate che i diodi D1 e D2 sono collegati in modo da fornire una funzione OR (alimentazione OR), ossia permettere di applicare tensione sia ad R3 che al contatto IN del CN2 simultaneamente evitando che la logica collegata a quest'ultimo venga sottoposta alla tensione portata sulla base del transistor T1 dalla pressione del pulsante o attraverso il contatto 4 del connettore CN1, che veicola un livello logico in grado di fare altrettanto.

UTILIZZO CON ARDUINO

Come accennato, una delle applicazioni del circuito proposto è l'accensione di una scheda Arduino che governa un circuito di vario genere e il cui sketch è predisposto per portare a livello alto un Digital Output quando riceve alimentazione (giusto il tempo di avviare il firmware subito dopo la

ricezione dell'alimentazione...) e lasciarlo tornare a zero quando ha svolto il suo compito, ovvero si è verificato un evento quale può essere l'acquisizione dei dati da un sensore e l'invio su porta seriale o interfaccia wireless.

Per farvi vedere come può essere implementata la cosa vi proponiamo un breve sketch di test che prevede queste connessioni tra la nostra scheda latch e Arduino Uno, meglio descritte dalla foto in **Fig. 1** che ritrae l'insieme:

- OUT del latch verso il D2 di Arduino;
- IN del latch verso il D3 di Arduino.

Per effetto dello sketch proposto nel **Listato 1**, quando si preme il pulsante e il MOSFET porta alimentazione al pin Vin di Arduino Uno, dopo l'inizializzazione e la definizione dei pin digitali viene portato ad 1 logico il pin digitale D3 in modo che Arduino si mantenga acceso anche dopo il rilascio del P1 sulla scheda e in assenza di tensione tra il piedino 4 del CN1 e massa.

Attraverso l'uscita OUT del circuito latch, Arduino riceve su D2 il livello basso determinato dalla pressione del pulsante, quindi esegue una routine e al completamento delle operazioni descritte dalle istruzioni qui di seguito si spegne:

```
digitalWrite(LED, HIGH);  
delay(1000);  
digitalWrite(LED, LOW);  
delay(1000);
```

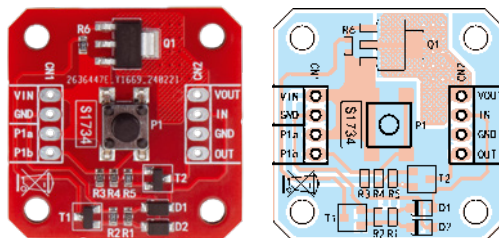
piano di MONTAGGIO

Elenco Componenti:

R1, R3, R4: 4.7 kohm (0603)
 R2, R5, R6: 100 kohm (0603)
 D1, D2: Diodo SM4007PL (SOD-123F)
 T1, T2: BC817 (SOT23)
 Q1: Mosfet ch-P NXMP6A17GTA (SOT223-3)
 P1: Microswitch SMD

Varie:

- Circuito Stampato S1734 (31x31 mm)



Le righe di codice soprastanti, nello specifico corrispondono all'accensione del LED on-board (quello connesso all'uscita D13 di Arduino UNO) e nel suo spegnimento dopo 1 secondo, quindi, trascorso un ulteriore secondo di tempo, il piedino D3 viene riportato a livello basso lasciando spegnere il circuito latch.

REALIZZAZIONE PRATICA

Bene, ciò detto passiamo agli aspetti costruttivi del progetto, partendo dal circuito stampato che abbiamo disegnato per accogliere i pochi componenti richiesti; il PCB per questo progetto è del tipo a doppia faccia e potete ottenerlo per fotoincisione a partire dalle tracce lato rame disponibili per il download sul nostro sito web ei.futuranet.it, oppure (se non avete l'attrezzatura o la voglia di fare da voi) richiedendolo tramite il servizio di realizzazione PCB di Futura Elettronica accessibile on-line all'indirizzo <https://ftrlink.it/pcb>.

I componenti utilizzati sono tutti per montaggio superficiale, ragion per cui vi occorre un saldatore a punta sottile, del filo di lega saldante di piccolo diametro (0,5 mm sarebbe l'ideale) una lente d'ingrandimento e una pinzetta per posizionare i componenti e magari della pasta flussante per agevolare la saldatura.

Una volta ottenuto il vostro circuito stampato potete disporvi i componenti occorrenti a partire dalle resistenze e dai diodi al silicio, passando poi al pulsante miniatura P1, ai due transistor bipolari e al MOSFET; per l'orientamento dei componenti polarizzati, ossia diodi e transistor, seguite l'indicazione fornita dal piano di montaggio che trovate in queste pagine.

Completate il montaggio del circuito inserendo e

saldando nelle rispettive piazzole le file di pin-strip che costituiranno i connettori CN1 e CN2; una volta verificata l'esattezza del montaggio, il vostro circuito latch di alimentazione è pronto per l'impiego, giacché non richiede alcuna impostazione o taratura preliminare.

CONCLUSIONI

In questo articolo vi abbiamo proposto un circuito tanto semplice quanto utile, soprattutto nell'elettronica dei giorni nostri, dove il contenimento dei consumi energetici e la possibilità di gestire interi rami circuitali, dispositivi e sistemi con livelli logici e dispositivi allo stato solido sono caratteristiche determinanti.

Con due transistor bipolari, altrettanti diodi e un MOSFET, più una manciata di componenti passivi, permette di gestire l'accensione e lo spegnimento di ogni genere di circuito logico implementando la funzione di autoritenuta e anche di comando retriggevole.

Siamo certi che saprete impiegarlo al meglio nei vostri progetti: buon lavoro!



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
FT1734M	Latch di alimentazione elettronico	5,60

IL MATERIALE VA RICHIESTO A:

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



Lampade da tavolo per ogni esigenza!

A LED, con lente, con ricarica wireless...

Lampada LED con lente d'ingrandimento



€94,00

Cod. TP246

- » Lente d'ingrandimento: 8 diottrie
- » Alimentazione: 220 VAC - 240 VAC / 50 Hz
- » Consumo: 9 W
- » Flusso luminoso: 685 lumen
- » Temperatura colore: 6000-7000 K
- » Montaggio: morsetto metallico per il fissaggio al banco

Lampada da tavolo a LED con lente d'ingrandimento



€39,00

Cod. TP206

- » Lente d'ingrandimento: 3 + 12 diottrie
- » Ingrandimento: 1,75 x + 4 x
- » Alimentazione: 230 VAC / 50 Hz - 5,5 W
- » Flusso luminoso: 400 lumen
- » Temperatura colore: 6000-7000 K
- » Diametro lente: 85 mm
- » Lunghezza braccio di supporto: 250 mm

Lampada LED con lente d'ingrandimento



€72,00

Cod. TP197

- » Lente d'ingrandimento: 3 diottrie
- » Flusso luminoso: 650 lumen
- » Temperatura colore: 6000K-7000K
- » Alimentazione: 230 VAC / 50 H
- » Consumo: 9 W
- » Montaggio: morsetto metallico per il fissaggio al banco

DISPONIBILE ANCHE CON
FLUSSO LUMINOSO DI 700 LUMEN
COD. TP193 - € 94,00

Lampada da tavolo a LED con lente d'ingrandimento



€49,50

Cod. TP248

- » Lente d'ingrandimento: 3 + 12 diottrie
- » Ingrandimento: 1,75 x + 4 x
- » Alimentazione: 230 VAC / 50 Hz - 6,5 W
- » Flusso luminoso: 520 lumen
- » Temperatura colore: 6500 K
- » Diametro lente: 90 mm
- » Lunghezza braccio di supporto: 290 mm

Lampada LED touch dimmerabile con ricarica wireless per smartphone



€34,00

Cod. EFL60

- » Temperatura colore: 2800 K - 3400 K - 4500 K - 5600 K
- » Flusso luminoso: 250 lumen
- » Alimentazione: 5 VDC / 2,5 A - a corredo cavo USB e adattatore da rete 220 VAC (opzionale)
- » Consumo: 5 watt
- » Ricarica wireless: 5 V / 1 A
- » Dimensioni (mm): 126x45x440

Lampada LED professionale con lente d'ingrandimento



€134,00

Cod. TP243

- » Lente d'ingrandimento: 3 diottrie
- » Flusso luminoso: 850 lumen
- » Temperatura colore: 6500 K
- » Fissaggio: morsetto da tavolo
- » Lunghezza del braccio: 790 mm
- » Consumo: da 0,5 a 12 W
- » Alimentazione: 230 VAC / 50 Hz - 5,5 W
- » Montaggio: morsetto metallico per il fissaggio al banco

Lampada professionale a LED dimmerabile - 880 lumen



€98,00

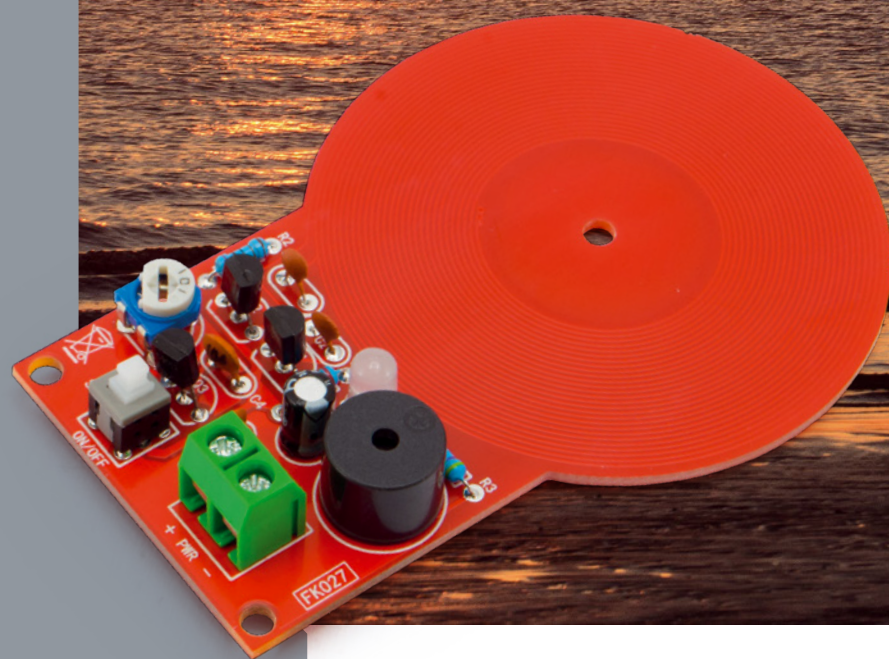
Cod. TP233

- » Alimentazione: 220 VAC - 240 VAC / 50 Hz
- » Consumo: da 1,5 a 14,5 W (dipende dalla luminosità)
- » Flusso luminoso: 880 lumen
- » Temperatura colore: 6000-7000 K
- » Montaggio: morsetto metallico per il fissaggio al banco
- » Dimensioni (mm): 450 x 80 x 400 (braccio)

Prezzi IVA inclusa.

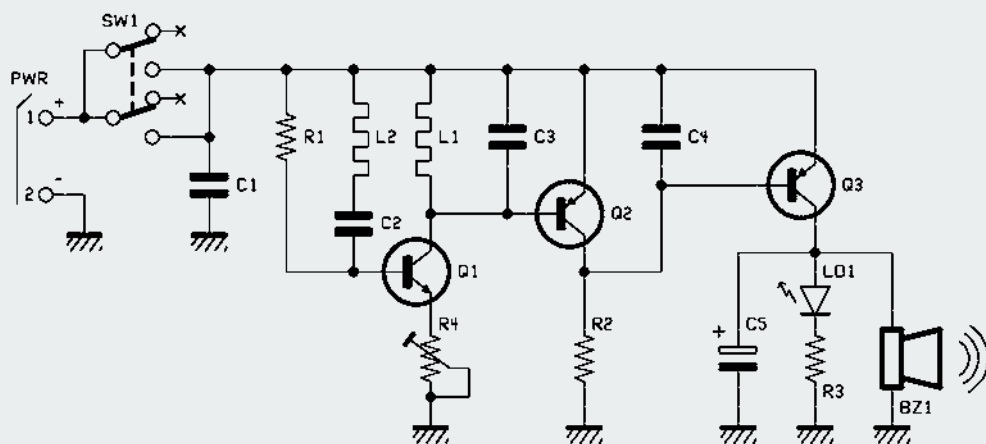
MINI METAL DETECTOR

di DAVIDE SCULLINO



Cercametalli di piccole dimensioni ideale sia per ritrovare monete e oggetti metallici nella terra e nella sabbia, sia per rilevare corpi metallici nei vestiti e nelle borse o tubi di ferro nei muri fino a 6 cm.

Li vediamo entrando in banca, in posta, andando in aeroporto o in tribunale ed anche semplicemente al mare, fuori stagione, in mano a qualcuno in cerca di oggetti metallici sotto la sabbia, per esempio monete, gioielli e quant'altro sia stato perduto in estate dai bagnanti; li abbiamo probabilmente visti nei negozi del fai da te oppure utilizzati in casa prima di piantare un chiodo, per capire se nel muro si trovano dei tubi. Si tratta dei metal detector o cercametalli che dir si voglia. Con il termine cercametalli si intende, in generale, qualsiasi apparato (tipicamente elettronico) capace di rilevare la presenza di metallo, sia ferromagnetico che diamagnetico o paramagnetico, a una certa distanza o all'interno di un varco a portale o di altro genere, sebbene tipicamente si tenda a

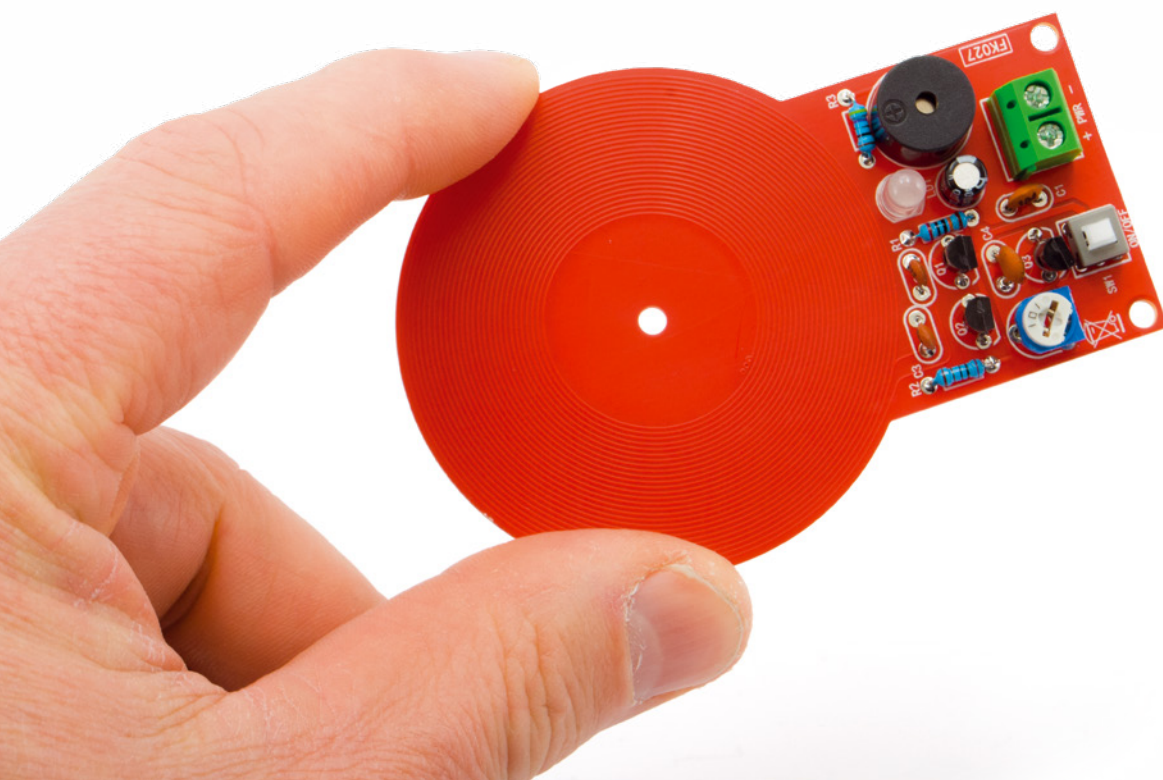


distinguere tali apparati in cercametalli e metal detector: con il primo nome si usa chiamare i dispositivi con cui si fa prospezione, ovvero si va alla ricerca di metalli preziosi, gioielli ed effetti personali, reperti archeologici nel sottosuolo, ma anche di tubature metalliche a muro o nel pavimento, mentre il termine metal detector generalmente identifica i dispositivi finalizzati alla ricerca di armi addosso alle persone o nei bagagli, di bombe sotterrate ed anche di protesi o placche metalliche nel corpo ai fini della preparazione ad esami come la Risonanza Magnetica Nucleare. In queste pagine vogliamo proporvi la realizzazione di un cercametalli palmare funzionante a pile e utilizzabile per verificare che in una parete dove volete piantare un chiodo o un tassello non vi siano tubazioni o cercare un oggetto metallico perduto

nella terra in giardino. Nulla vieta di utilizzarlo per rilevare la presenza di parti metalliche addosso alle persone, sebbene non si tratti di un vero cercametalli per addetti alla sicurezza come quelli utilizzati negli aeroporti o nei tribunali.

SCHEMA ELETTRICO

Entriamo subito nel vivo del progetto iniziando con l'analisi del relativo schema elettrico, illustrato in queste pagine; il circuito è davvero semplice, quasi banale, sebbene il suo principio di funzionamento sia abbastanza raffinato; l'hardware consta di tre transistor, il primo dei quali è un oscillatore a retroazione cui segue un amplificatore di potenza con filtro capacitivo che incorpora un interruttore statico. L'oscillatore è di tipo Meissner e funziona intorno al transistor NPN Q1, al cui collettore è applicato



Le dimensioni del cercametalli sono molto contenute, tanto da renderlo tascabile.

un circuito antirisonante (formato dall'induttanza L1 e C3) accordato alla frequenza di oscillazione, la cui bobina costituisce anche l'avvolgimento primario di un trasformatore in aria; il secondario è la bobina L2, che si trova in serie al condensatore C2, necessario a disaccoppiare la componente continua di polarizzazione a riposo della base del transistor (se non ci fosse, la base del Q1 sarebbe cortocircuitata con il positivo di alimentazione e l'oscillatore non funzionerebbe) ma dimensionato per farsi attraversare dal segnale alla frequenza di lavoro dell'oscillatore, presentando reattanza capacitiva trascurabile e ininfluente sul funzionamento del transistor. Per l'esattezza, C2 forma idealmente con la L2 un circuito risonante serie accordato alla stessa frequenza di quello risonante parallelo formato da L1 e C3.

Le due bobine anzidette, per facilitarvi la realizzazione, le abbiamo realizzate direttamente sul circuito stampato, nel senso che sono due piste a spirale concentriche e opportunamente dimensionate.

L'oscillatore di Meissner rientra nella categoria dei circuiti oscillatori a trasformatore, nella quale la retrocessione del segnale necessario a innescare l'oscillazione avviene mediante un trasformatore che accoppia il circuito di uscita con quello di ingresso. Lo sfasamento di 180° viene determinato dal fatto che il trasformatore restituisce al secondario una tensione indotta che è sempre in opposizione di fase rispetto a quella che alimenta il primario (quindi, 180° esatti). Nell'oscillatore a trasformatore la frequenza dipende dalle caratteristiche del transistor e dall'induttanza degli avvolgimenti, ovvero dal ritardo che il trasformatore introduce nella pratica a causa delle proprie componenti parassite e che non è sempre 180° come dovrebbe essere in teoria. Nell'oscillatore di Meissner il trasformatore retrocede al primario una tensione che, per come sono realizzati gli avvolgimenti (le due spirali sono affiancate e la fine dell'L1 si trova in corrispondenza dell'inizio dell'L2 e viceversa), è in opposizione di fase e tale da innescare l'oscillazione: in pratica quando la tensione ai capi della L1 cresce perché il potenziale sul collettore diminuisce a seguito della maggiore conduzione del BJT (aumento della corrente di collettore) cresce anche la differenza di potenziale ai capi della L1 e il potenziale di base viene incrementato determinando un ulteriore aumento della corrente di collettore e un abbassamento del potenziale di collettore. Invece, quando il potenziale di collettore cresce (perché il transistor va all'interdizione), sulla

CARATTERISTICHE TECNICHE

❖ Tensione di alimentazione: 9÷12V ❖ Corrente assorbita: 50 mA ❖ Funzionamento: a variazione d'induttanza	❖ Rilevamento: tutti i metalli ❖ Segnalazione rilevamento: acustica e visiva
---	--

base giunge una tensione che spinge l'interdizione. Insomma, alla base giunge un segnale in fase con quello presente sul collettore e ciò innesca l'oscillazione, al contrario di quanto avverrebbe utilizzando il circuito da amplificatore, dove la retroazione tende a contrastare l'incremento di ampiezza del segnale d'uscita.

Il funzionamento si può spiegare immaginando che nel rumore termico generato dai semiconduttori e in quello elettrico è presente sicuramente una frequenza tale che, amplificata dal circuito di collettore, riesca a far retrocedere alla base una tensione di ampiezza sufficiente a innescare e mantenere l'oscillazione. Tale condizione corrisponde a un prodotto tra tasso di retroazione e guadagno dello stadio a transistor uguale a 1, in condizioni di retrocessione di un segnale che spinga il transistor ad amplificare invece che a ridurre il guadagno.

La frequenza d'innescò è quella di accordo del circuito antirisonante posto sul collettore quindi dipende dai valori di L1 e C3; siccome l'oscillazione si può innescare quando il prodotto tra segnale di retroazione positiva e guadagno è uguale a 1, è stato inserito il trimmer R3, che consente, attraverso la variazione della resistenza di emettitore, di raggiungere il guadagno desiderato, occorrente all'innescò dell'oscillatore.

Più esattamente, il trimmer permette di regolare il guadagno per ottenere l'innescò dell'oscillazione, ma anche di controllare la sensibilità del metal-detector attraverso il raggiungimento della massima ampiezza della tensione in oscillazione. La frequenza di lavoro, ossia quella alla quale oscilla il circuito, si determina precisamente grazie al fatto che il primario del trasformatore



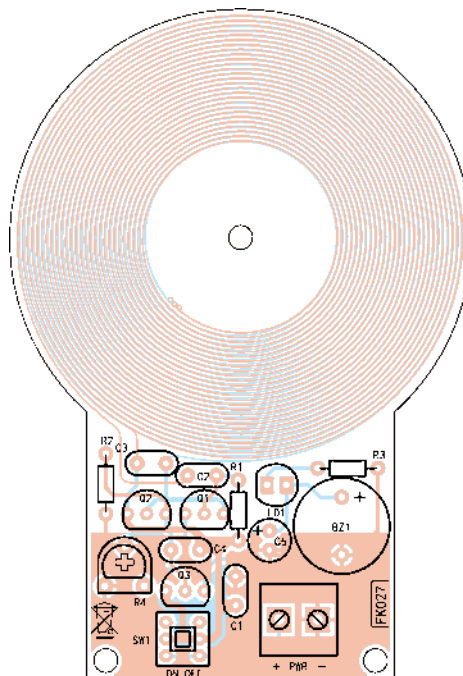
Elenco Componenti:

R1 200 kohm
R2 2 kohm
R3 470 ohm
R4 Trimmer 1 kohm MO
C1, C4 100 nF ceramico
C2, C3 2,2 nF ceramico
C5 100 µF 25 VL elettrolitico
Q1 9018

Q2, Q3 9012
BZ1 Buzzer con elettronica 5V
LD1 LED 5 mm Rosso
SW1 Interruttore a Pulsante DPDT

Varie:

- Morsetto 2 poli passo 2,54 mm
- Circuito Stampato FK027 (61x89 mm)



forma, con il condensatore che gli si trova in parallelo, un circuito accordato antirisonante; quest'ultimo ha la caratteristica di presentare un'impedenza elevatissima -teoricamente infinita- in corrispondenza della frequenza di risonanza (quella alla quale la reattanza capacitiva eguaglia quella induttiva, almeno nel modulo) e molto bassa alle altre frequenze. In virtù di ciò, il trasformatore trasferisce alla base del transistor un segnale abbastanza ampio da innescare l'oscillazione praticamente alla sola frequenza di risonanza del circuito LC.

Si può quindi affermare, con buona approssimazione, che la frequenza di lavoro vale:

$$f_0 = \frac{1}{2 \pi \sqrt{L \times C}}$$

Dunque, l'oscillatore emette teoricamente un segnale sinusoidale a frequenza uguale a quella di risonanza del bipolo LC.

Notate che nel circuito le due resistenze di sinistra servono alla polarizzazione in continua della base e il condensatore provvede a cortocircuitarle in presenza di segnale.

L'oscillatore di Meissner esiste in due varianti, la prima delle quali è quella appena descritta; l'altra ha il circuito accordato sul secondario (base) invece che sul primario (collettore).

La frequenza di oscillazione del caso si calcola con una forma analoga a quella appena esposta, con la sola differenza che al posto di L va il valore di induttanza del secondario, ossia della bobina che sta sulla base del transistor.

Fin qui abbiamo spiegato come funziona l'oscillatore ma senza esporne l'utilità ai fini del

rilevamento dei metalli; ora è giunto il momento di spiegare qual è la sua funzione: il trasformatore di cui L1 e L2 sono primario e secondario è "avvolto" in aria, ovvero senza alcun nucleo ferromagnetico che incrementi il flusso magnetico trasferito, quindi se al circuito stampato del nostro metal detector viene avvicinato un pezzo di metallo ferromagnetico, la permeabilità sale notevolmente e cambia l'induttanza di L1 ed L2.

Tale condizione altera la frequenza di lavoro dell'oscillatore e questo fenomeno viene sfruttato dalla porzione di circuito che segue; per l'esattezza, Q2 è configurato come amplificatore invertente che amplifica il segnale di collettore del transistor costituente l'oscillatore di Meissner e il condensatore C4, insieme alla resistenza di collettore R2, forma un filtro passa-basso del prim'ordine che taglia la frequenza nominale a vuoto dell'oscillatore, mentre lascia passare una componente filtrata, continua, quando la frequenza scende al disotto di essa, come nel caso in cui al circuito stampato viene avvicinato un oggetto metallico. Notate che la frequenza di lavoro viene alterata anche dalla presenza di metalli non ferromagnetici, in quanto essendo elettricamente conduttivi, le tensioni indotte in essi dalla presenza del campo elettromagnetico irradiato dalla bobina L1 vengono cortocircuitate, determinando un carico nell'oscillatore (sotto forma di maggiore assorbimento dal transistor

Q1), tale da determinare un abbassamento della frequenza di oscillazione. In questo caso il rilevamento avviene per effetto delle correnti parassite (le cosiddette "eddy currents") determinate dal cortocircuito della tensione indotta che si verifica nei materiali conduttori. In tutti i casi, più la frequenza sul collettore del Q1 scende, più cresce l'ampiezza della tensione ai capi del condensatore di filtro C4 (perché diminuisce la caduta sulla R4), il che porta gradualmente il transistor Q3 (PNP anch'esso...) a condurre sempre più (perché cresce la differenza di potenziale tra emettitore e base), pulsando con il proprio collettore sul condensatore elettrolitico C5, che filtra ulteriormente il segnale residuo variabile e con la tensione continua che viene così ricavata alimenta il cicalino BZ1; quest'ultimo è del tipo con oscillatore integrato, pertanto ogni volta che riceve l'alimentazione emette una nota acustica generata da un circuito interno. Il suono emesso dal cicalino piezo è tanto più intenso quanto più la frequenza di oscillazione si abbassa a causa della vicinanza del metallo o della consistenza del metallo stesso, perché ciò fa alzare la tensione che Q3 fornisce al BZ1.

Oltre all'avviso acustico, la conduzione del Q3 determina l'accensione del LED LD1. Concludiamo l'analisi dello schema elettrico con l'alimentazione dell'intero circuito, che si applica ai punti + e - PWR e da esso arriva al doppio deviatore SW1, che utilizziamo come interruttore per

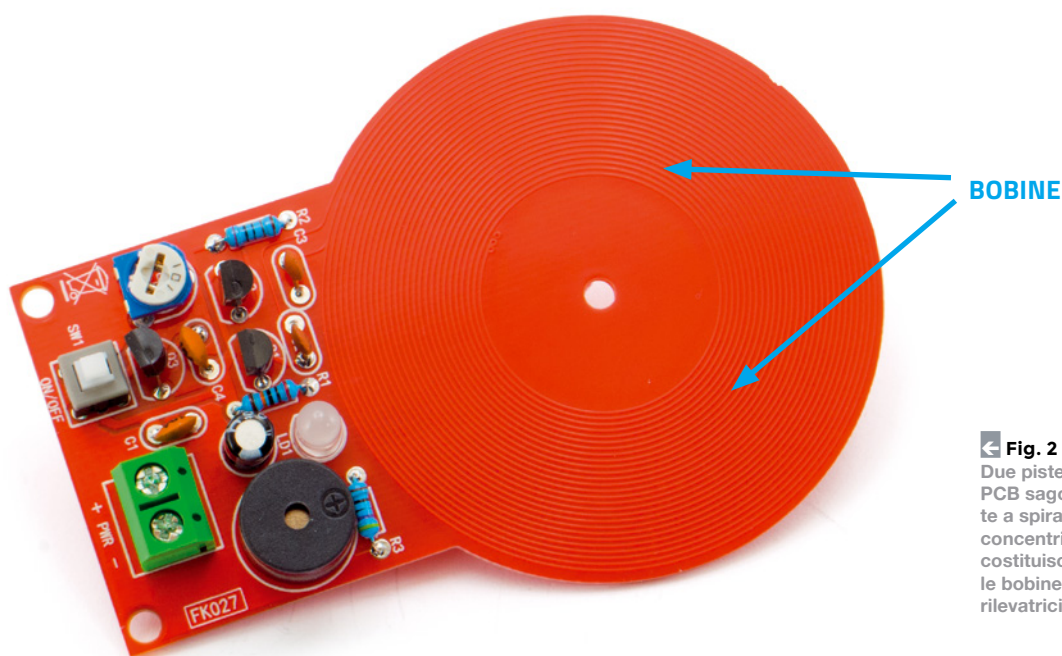


Fig. 2
Due piste del PCB sagomate a spirale e concentriche, costituiscono le bobine rilevatrici.

Il principio di funzionamento del metal detector è la reazione che il campo magnetico generato da una bobina elettrica, utilizzata nell'esplorazione del terreno, del muro o di una persona da analizzare, presenta quando in esso viene introdotto qualcosa di ferromagnetico, paramagnetico o diamagnetico purché elettricamente conduttivo. Il rilevamento può avvenire perché il materiale introdotto:

- modifica la permeabilità assoluta del mezzo in cui si crea il flusso magnetico indotto e quindi varia l'induttanza della bobina esploratrice, la quale, essendo alla base di un oscillatore, provoca una deviazione della frequenza generata;
- elevando la permeabilità, consente un più alto flusso magnetico; è il caso degli apparati basati su trasformatore, nei quali il materiale captato modifica l'accoppiamento

magnetico tra due solenoidi;

- se è conduttore, diventa sede di correnti parassite superficiali o meno, che, a loro volta, provocano un campo magnetico contrastante quello del metal-detector.

Il primo caso riguarda i cercametri a variazione di induttanza, mentre quelli a variazione del flusso sono basati su speciali trasformatori realizzati con una bobina alimentata a corrente variabile, che perciò induce in un secondo solenoide un flusso magnetico e determina ai suoi capi una tensione indotta. Ma i metal detector funzionano anche e soprattutto sfruttando l'effetto delle correnti parassite; questo spiega come mai si possono rintracciare anche metalli quali oro e argento, che non manifestano effetti magnetici e in generale tutti i metalli: ferromagnetici, diamagnetici e paramagnetici, sfruttano proprio le correnti parassite.

I cercametri per materiali ferromagnetici funzionano prettamente a variazione di induttanza, ovvero di permeabilità magnetica del mezzo in cui vengono irradiate le onde elettromagnetiche. Quando un corpo ferromagnetico viene a trovarsi nel campo generato da un induttore avvolto in aria, può far salire fortemente la permeabilità, tanto da determinare una netta variazione di frequenza (se la bobina è parte di un oscillatore LC) ovvero il flusso tra tale induttore e una seconda bobina che va da secondario (è il caso dei sistemi a trasformatore). Lo stesso principio può essere utilizzato per rilevare i materiali paramagnetici, anche se è meglio ricorrere all'azione dovuta a quelle che in Fisica sono chiamate correnti parassite di Foucault o correnti superficiali. Il fenomeno si spiega considerando che in un corpo immerso nelle linee di forza di un campo

magnetico variabile, quale quello generato dalla bobina di ricerca del metal-detector, viene indotta una tensione che, essendo il metallo un conduttore, determina una corrente. La corrente, che rappresenta un fenomeno parassita, è a sua volta causa di un campo magnetico che interagisce con quello che l'ha generata. La chiusura dei circuiti virtuali all'interno della massa metallica o conduttrice, dalla superficie all'interno, fa sì che il metallo o conduttore investito dal campo di un metal detector reagisca generando un proprio campo elettromagnetico. Questo fenomeno porta a sottrarre energia al campo che l'ha generato, perché la corrente in cortocircuito di una spira immersa in un campo magnetico variabile dipende da quella che alimenta la bobina responsabile del campo magnetico. Anzi, per l'esattezza, l'energia elettrica sviluppata dalle correnti parassite viene

accendere e spegnere il circuito senza dover di volta in volta sconnettere manualmente la fonte di alimentazione, che nello specifico può essere tranquillamente una pila a secco da 9 volt, visto che l'assorbimento del circuito è abbastanza contenuto da permetterlo e fornire lunga autonomia. La linea di uscita dal doppio deviatore viene filtrata dal condensatore C1 (che abbate i disturbi impulsivi eventualmente captati nell'ambiente dai fili dell'alimentazione) e quindi raggiunge il resto del circuito.

REALIZZAZIONE PRATICA

Passiamo adesso dalla teoria alla pratica, spiegando come si costruisce il metal detector: la prima operazione da compiere è preparare il circuito stampato, che è del tipo a doppia faccia e che raccomandiamo di ottenere a partire dalle tracce lato rame pubblicate (e scaricabili) sul nostro sito elettronica.in.it insieme agli altri file del progetto; scaricati i file, stampateli su carta da lucido o acetato e otterrete le pellicole per procedere con la fotoincisione impressionando una faccia alla volta. Inciso e forato il circuito stampato,

disponetevi i pochi componenti richiesti a partire da resistenze e condensatori (dando la precedenza a quelli non polarizzati) e poi passate al trimmer e ai transistor; stagiate bene da entrambe le facce i reofori dei terminali che hanno piazzole in comune su entrambi i lati, così da realizzare le interconnessioni (vie) tra le due ramature. Terminate il montaggio con il piccolo deviatore a pulsante da c.s. ed il cicalino, oltre che la morsettiera che utilizzerete per alimentare il metal-detector.

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

Una volta completate le saldature, tagliate le eccedenze dei terminali e verificato che tutto sia a posto, potete già utilizzare il vostro metal detector, visto che l'unica taratura, ossia quella del trimmer, si esegue "sul campo", cioè provando ad avvicinare il circuito (alimentato con una pila da 9V) a delle parti metalliche e ruotando il cursore fin quando non si sente il suono uscire dal cicalino.

In vero il trimmer si può registrare anche con degli strumenti e, nello specifico, con un oscilloscopio,

prelevata, per accoppiamento magnetico, dal circuito della bobina inducente. Ciò vuol dire che la presenza di un corpo elettricamente conduttore richiede un maggiore assorbimento di elettricità dal circuito che pilota la bobina di ricerca del metal-detector; rilevando la variazione di consumo, si rileva la presenza. Rilevare l'assorbimento nei metal-detector permette la ricerca degli oggetti composti da elementi paramagnetici; anzi, i cercametri progettati allo scopo sono in grado di rilevare i due effetti che si originano facendo entrare un corpo paramagnetico nel campo elettromagnetico prodotto dalla bobina di ricerca:

- la modifica della permeabilità assoluta del mezzo in cui si crea il flusso magnetico indotto e quindi la variazione dell'induttanza della bobina esploratrice;

- la variazione di assorbimento dovuta all'energia sottratta dallo scorrimento delle correnti parassite.

Teoricamente si potrebbe fare altrettanto anche nella ricerca dei materiali ferromagnetici, tuttavia in questi la variazione di permeabilità è preponderante e di per sé sufficiente a determinare indicazioni facilmente discriminabili dai falsi allarmi dovuti ai disturbi. In linea generale gli apparecchi che rilevano le variazioni di assorbimento o di frequenza dovute all'azione delle correnti parassite sul suolo o negli oggetti sottostanti, sono adatti a rintracciare qualsiasi massa che risulti elettricamente conduttiva, risultando tanto più sensibili quanto più bassa è la loro resistenza elettrica. I metal-detector che rilevano le variazioni di frequenza (dovute alla variazione di induttanza della

bobina inducente) o di corrente in un avvolgimento indotto sono invece prettamente dedicati alla ricerca di materiali ferromagnetici e paramagnetici.

Le tecnologie utilizzate nei metal detector odierni sono:

- BFO; rilevano masse ferrose, dato che hanno una bobina la cui induttanza varia in funzione della permeabilità magnetica della zona esplorata; rilevano comunque anche oggetti conduttivi;
- TR; quelli che sfruttano il principio del trasformatore rilevano solo oggetti ferromagnetici, perché la loro presenza aumenta l'intensità del flusso e quindi della corrente prelevabile sulla bobina secondaria; gli IB e i TR a sintonia, sentono invece anche i conduttori e, purtroppo, l'effetto suolo, il quale è dovuto al flusso di correnti parassite nel terreno e all'effetto che producono, tale da falsare il rilevamento;

- PI; lavorano essenzialmente sul ferro e quasi per niente sui diamagnetici, in quanto si basano sul rilevamento della magnetizzazione residua degli oggetti investiti da un flusso; per questa ragione, possono rilevare oggetti ferromagnetici anche molto piccoli, ma ignorano i metalli preziosi;
- VLF; combinano l'effetto delle correnti parassite tipico degli elementi diamagnetici a quello della variazione di permeabilità magnetica caratteristico dei ferromagnetici e dei paramagnetici; sono insensibili all'effetto suolo perché, pur rilevando l'assorbimento dovuto alle correnti di Foucault, funzionano sentendo lo spostamento di fase provocato da esse e nelle versioni gestite da microprocessore o microcontrollore, scartano la risposta alle frequenze alle quali il suolo mineralizzato dà maggiore risalto.

collegando la sonda di quest'ultimo tra la massa (negativo di alimentazione) e il collettore del transistor Q1 e verificando quando, ruotando il cursore del trimmer, si sviluppa la sinusoide corrispondente all'oscillazione (se il circuito non oscilla il collettore avrà un potenziale costante...) e cercando di ottenere la massima ampiezza. Peraltro con l'oscilloscopio potete anche verificare ciò che accade al segnale dell'oscillatore, ossia alla tensione variabile localizzata sul collettore del transistor Q1 quando il metal detector viene avvicinato a un pezzo di metallo. Una volta regolato il circuito è il caso di trovare un contenitore per racchiuderlo, che dovrà necessariamente essere non metallico (va bene un classico involucro in plastica) per evitare che venga rilevato esso stesso, determinando false segnalazioni; il circuito stampato dovrà essere applicato col lato saldature aderente a un pannello del contenitore, in modo da limitare il più possibile la distanza tra la bobina e le parti da analizzare, a tutto vantaggio della sensibilità del metal detector. La pila dovrà essere posta abbastanza lontano dalle bobine, ovvero dalla parte del

circuito stampato dove si trova la morsettiera di alimentazione, pena l'interferenza con il rilevamento dei metalli.

Bene, con questo riteniamo di aver concluso con i suggerimenti pratici e vi lasciamo alla sperimentazione del vostro nuovo cercametri palmare, dal quale, siamo certi, avrete molte soddisfazioni e occasioni per sorprendervi vedendo il comportamento che i vari oggetti determinano. □

COSA OCCORRE ?		
Codice	Descrizione	Prezzo € IVA inclusa
FK027	Cercametri in scatola di montaggio	7,50
9V6LP3146	Batteria Alcalina VARTA Longlife 9 V	2,80
IL MATERIALE VA RICHIESTO A:		
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA) Tel: 0331-799775 - www.futurashop.it		



Non lasciare la tua casa incustodita!

Allarme evoluto WiFi-GSM/GPRS controllabile da remoto



€119,00

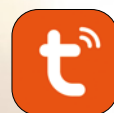
Cod. FR783



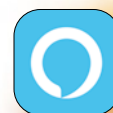
Controllabile con:



Smart Life



Tuya Smart



Amazon Alexa



Google Home

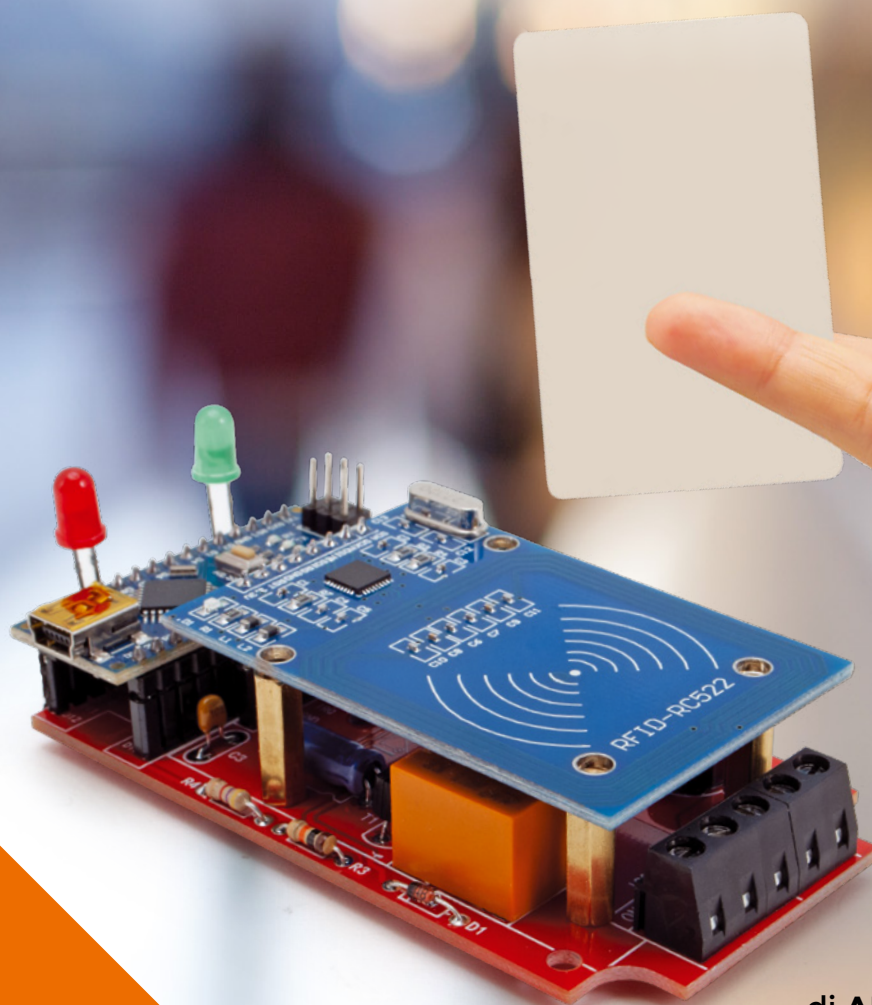
Centralina antifurto wireless programmabile, controllabile da remoto tramite Smartphone sfruttando la rete GSM/GPRS o Wi-Fi e direttamente con Amazon Alexa o Google Home. Le App "Smart Life" o "Tuya Smart" permettono di configurare e gestire il funzionamento dell'allarme in ogni suo dettaglio. È possibile aggiungere sensori ed accessori opzionali per coprire fino a 99 zone. La confezione comprende: la centralina, due radiocomandi, un sensore wireless per porte/finestre, un sensore PIR wireless, due chiavi TAG RFID, la batteria di backup e l'alimentatore di rete. Manuale di utilizzo in italiano.

Accessori opzionali:

								
€32,00 Cod. FR783KEY	€34,00 Cod. FR783PIREXT	€14,00 Cod. FR783PIR	€11,00 Cod. FR783MAGN	€5,00 Cod. FR783TAG	€19,00 Cod. FR783SMOKE	€9,00 Cod. FR783RAD	€29,00 Cod. FR783SIR	€72,00 Cod. FR783SIREXT

Prezzi IVA inclusa

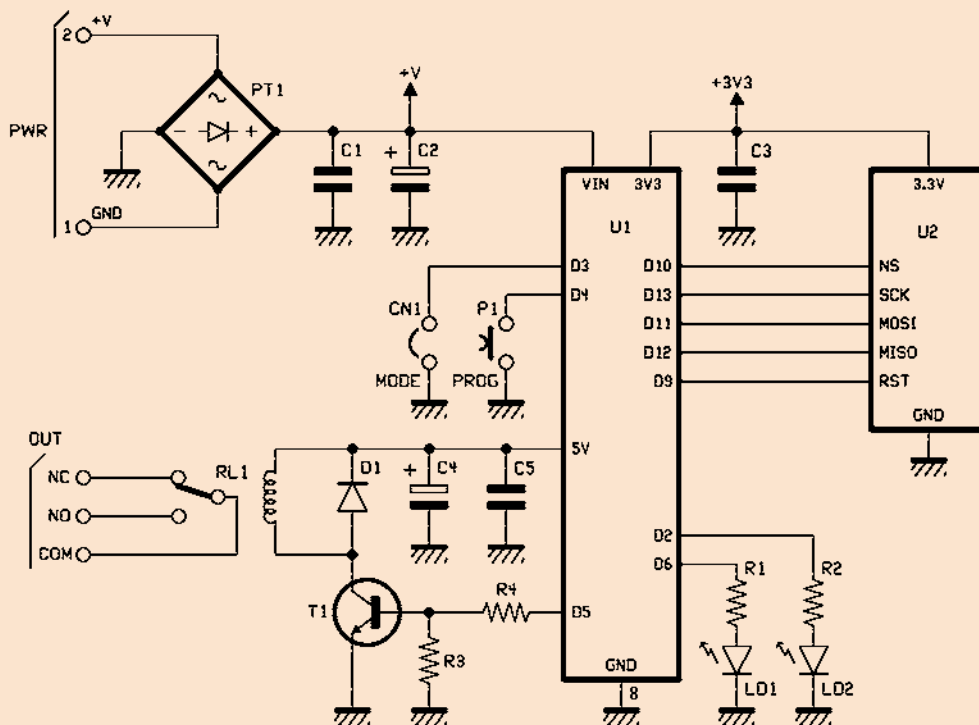
SERRATURA **RFID**



di **ALESSANDRO SOTTOCORNOLA**

Operante sulla frequenza standard di 13,56 MHz, viene governato da una piccola Arduino Nano per aprire e chiudere un relé in base al riconoscimento dei tag avvicinati al modulo RFID.

Il controllo accessi è una delle applicazioni più in voga da qualche decina d'anni, tanto che è stato esteso ormai in tutti gli ambiti della vita quotidiana: dall'entrata riservata al personale in azienda, all'accesso dei soci di club e frequentatori di piscine e palestre, fino ad arrivare al controllo dell'apertura del cancello di casa o all'accesso alle macchinette per caffè e bevande. Indubbiamente la tecnologia più apprezzata per il controllo accessi è quella contactless e nello specifico l'RFID, perché consente di identificarsi semplicemente avvicinando un tag che può essere un portachiavi o una tessera, senza contatto fisico e senza bisogno di introdurre card a banda magnetica o chip, con tutti i vantaggi che ne derivano in termini di affidabilità (i tag RFID non si smagnetizzano) e di durata (non c'è l'attrito implicato dai lettori di card magnetiche o ChipCard).



Proprio nell'ambito dell'RFID è nato l'apprezzatissimo NFC (Near Field Communication) che se vogliamo ne è un'estensione a dispositivi come tablet e smartphone (ma anche Smart-Watch) e sempre con l'RFID sono state sviluppate numerose varianti distinte per frequenza di lavoro e funzioni supportate, tra le quali l'antitaccheggio o l'identificazione e gestione automatizzata delle merci di magazzino.

Sul tema abbiamo già lavorato in passato progettando e proponendovi serrature elettroniche dotate di lettore RFID, come ad esempio il popolare ID-12, molto apprezzato perché contenente tutto l'occorrente per programmare e leggere tag RFID passivi. Vogliamo, adesso, tornare sull'argomento, proponendovi una serratura elettronica provvista di un relé che scatta quando il tag avvicinato all'antenna viene riconosciuto.

Con lo stesso circuito è anche possibile programmare i tag da utilizzare poi con la serratura, abbinandoli ad essa; spiegheremo più avanti come svolgere le procedure del caso. Ora dedichiamoci alla consueta analisi circuitale, per vedere in cosa consiste il progetto.

SCHEMA ELETTRICO

Per quanto complesso, l'insieme è reso semplice dall'adozione di due schede elettroniche capaci di

assolvere ai due compiti principali, ossia gestire la lettura e scrittura dei tag RFID (a questo compito è deputato l'apposito modulo RFID) e governare l'attività del relé sulla base delle risultanze dell'attività del modulo RFID.

Il modulo RFID è un VMA405N e incorpora l'antenna irradiante/ricevente, l'oscillatore che genera il segnale di pilotaggio della bobina d'antenna, il circuito discriminatore della pulsazione del segnale e un microcontrollore provvisto di interfaccia dati a protocollo SPI con il quale dialoga con la board Arduino Nano. Il modulo RFID da noi utilizzato semplifica notevolmente l'hardware e consente di leggere e scrivere schede RFID di tipo MiFare; prende posto sul circuito stampato sul quale viene ospitata anche la board Arduino, in modo da ottenere un insieme compatto e non avere fili volanti. Il controller su cui si basa è l'MFRC522 della NXP, un chip di gestione della lettura e scrittura di tag per la comunicazione contactless alla frequenza di 13,56 MHz: nella **Fig. 1** ne vedete lo schema a blocchi. Il trasmettitore interno dell'MFRC522 è in grado di pilotare un'antenna di lettura/scrittura progettata per comunicare con tessere e transponder ISO/IEC 14443 A/MIFARE senza richiedere circuiti aggiuntivi come driver per la bobina. Lo stadio ricevitore fornisce un'implementazione robusta ed efficiente per demodulare e decodifi-

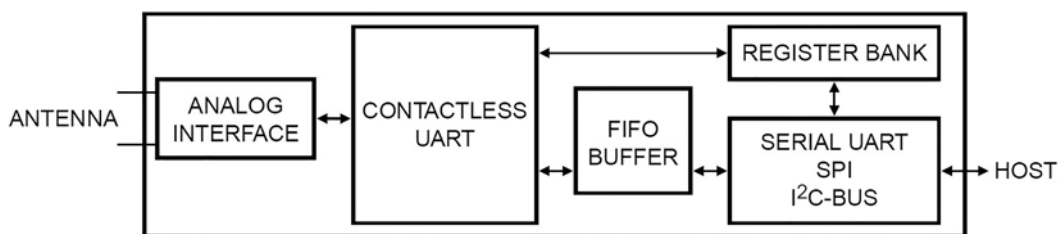


Fig. 1
Schema
a blocchi
semplificato
dell'MFRC522.

care segnali da tag e schede compatibili ISO/IEC 14443 A/MiFare. Il modulo digitale gestisce l'intero framing ISO/IEC 14443 A e funzionalità di rilevamento errori (parità e CRC) oltre a supportare la comunicazione RF a velocità di trasferimento fino a 848 kBaud in entrambe le direzioni. Con riferimento allo schema elettrico, il modulo RFID è il componente siglato U2 e si connette tramite un apposito pin-strip a 8 poli (di cui uno non utilizzato e due per l'alimentazione) alla scheda, dalla quale prende i 3,3V per il proprio funzionamento, filtrati dal condensatore C3 e forniti dal regolatore interno alla Arduino Nano, siglata U1; quest'ultima si interfaccia al canale dati SPI (del quale assume il ruolo di unità Master) con il modulo RFID (Slave dell'SPI) secondo questa corrispondenza:

- D9 di Arduino gestisce il reset dell'SPI;
- D10 l'NS del modulo RFID;
- D11 di Arduino invia i dati al modulo attraverso la linea MOSI (Master Output, Slave Input);
- D12 di Arduino riceve i dati dal MISO (Master Input, Slave Output) del modulo;
- D13 di Arduino fornisce al modulo il clock per scandire la comunicazione seriale.

Il modulo RFID è quindi gestito completamente da Arduino Nano, che invia i comandi di scrittura e legge i dati, ponendosi in ascolto sulla linea MISO, inviati dal modulo stesso quando esso acquisisce un tag.

La comunicazione con il modulo RFID è gestita da Arduino Nano affinché acquisisca un tag quando questo viene avvicinato al lettore. In particolare, Arduino Nano comanda il modulo RFID tramite la linea MOSI affinché generi il proprio campo elettromagnetico, stimolando così il tag esposto alle linee di forza a inviare impulsi corrispondenti ai codici. Il processore del modulo discrimina questi impulsi, li converte in formato SPI e li invia tramite la linea MISO ad Arduino, affinché esso possa scriverli nella propria memoria dei tag abbinati in modo efficiente e sicuro.

Durante il normale funzionamento, il modulo RFID

viene interrogato ciclicamente da Arduino Nano per verificare se un tag è entrato nel suo campo d'azione e se i dati corrispondenti sono stati discriminati e letti. Quando ciò avviene, il firmware che gira su Arduino Nano verifica che il codice contenuto nella stringa dei dati su SPI corrisponda a uno di quelli in memoria (in EEPROM). In caso affermativo, interviene sul relé di uscita della scheda (RL1) secondo la modalità impostata dal ponticello MODE (CN1) connesso al pin digitale D3 (provvisto internamente di resistore di pull-up) di Arduino Nano.

La funzione del ponticello verrà descritta tra breve, ma possiamo anticipare che sono previste due modalità di funzionamento, contraddistinte da due livelli di sicurezza, per garantire un utilizzo versatile ed efficace del sistema.

In ogni caso per il relé è prevista la sola attivazione impulsiva (monostabile, quindi il relé scatta per un certo periodo al riconoscimento di un tag valido). In caso contrario ignora i dati e Arduino Nano ritorna in "ascolto" sulla linea MISO, ovvero in attesa che venga premuto il pulsante P1 che permette di entrare nella modalità di memorizzazione di una carta.

Il relé viene comandato da Arduino Nano portando a livello logico alto la propria linea D5, il cui stato logico polarizza, attraverso il partitore resistivo formato da R3 ed R4, la base del transistor T1, il cui collettore alimenta la bobina del RL1, in parallelo alla quale abbiamo inserito il diodo D1; la funzione di quest'ultimo è proteggere la giunzione base-collettore dell'NPN quando il transistor si interdice, perché sospendendo bruscamente la corrente alla bobina, questa genera un'extratensione inversa che danneggerebbe irrimediabilmente T1 cortocircuitandone definitivamente la giunzione base-collettore.

Del relé rendiamo disponibile l'intero scambio, ossia comune (C) normalmente aperto (NO) e contatto normalmente chiuso (NC) in modo da poter utilizzare il nostro circuito nella maniera più versatile possibile; per esempio con due contatti dello scambio potete chiudere la linea di

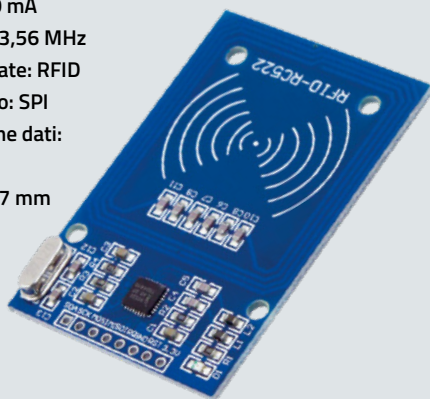
Per semplificare il circuito della serratura elettronica e sollevarvi dall'onere di tararlo o calibrarlo, ci siamo affidati a un modulo già pronto montato e collaudato, in grado di leggere e programmare tag RFID basato sullo standard MiFare. Il modulo, siglato VMA405N, è gestito da un chip controller MFRC522 ed ha le seguenti caratteristiche tecniche:

- tensione d'esercizio: 3,3 VCC
- corrente di lavoro: da 13 a 26 mA
- corrente di riposo: < 80 uA
- corrente di picco: < 30 mA
- frequenza di lavoro: 13,56 MHz
- tipi di schede supportate: RFID
- interfaccia / protocollo: SPI
- velocità di trasmissione dati: massimo 10 Mbit / s
- dimensioni: 66 x 40 x 7 mm

Grazie all'interfaccia a bus SPI di cui dispone, lo possiamo tranquillamente gestire con una scheda Arduino nano, per la quale abbiamo preparato un apposito sketch.

Il modulo RFID è proposto dall'immagine in questo riquadro.

Il controller MFRC522 su cui il modulo si basa supporta tutte le varianti del protocollo di identificazione RF MiFare, ossia MiFare Mini, MiFare 1K, MiFare 4K, MiFare Ultralight, MiFare DESFire EV1 e MiFare Plus RF.



alimentazione di un'elettroserratura o l'ingresso di comando di un tornello, ma anche attivare un segnalatore ottico o acustico. L'importante è che il circuito o utilizzatore da controllare funzioni a tensione continua o alternata fino a 230Vca ed assorba non più di 3 ampere.

Il relé e il relativo circuito di comando funzionano con la tensione continua di 5V prelevata dal pin 5V di Arduino Nano, che la fornisce attraverso il proprio regolatore interno a partire dalla tensione che gli viene fornita al pin Vin. Quest'ultimo è alimentato attraverso un alimentatore c.a./c.c. composto dal ponte raddrizzatore di Graetz integrato siglato PT1, il cui ingresso è collegato alla morsettiera PWR e la cui uscita, opportunamente livellata e filtrata dai condensatori C1 e C2, raggiunge proprio Vin di Arduino Nano e la relativa massa. Notate che il ponte a diodi ha una doppia funzione: consente l'alimentazione del circuito sia a corrente alternata (con una tensione di valore compreso fra 8 e 10V) sia a corrente continua: in questo caso offre il vantaggio di poter applicare l'alimentazione al circuito senza curarsi della polarità.

Chiudiamo la descrizione con i due LED, uno rosso e uno verde, che Arduino Nano pilota con le linee digitali D2 (diodo verde) e D6 (LED rosso), inizializzate come uscite; i diodi luminosi servono a fornire le segnalazioni sullo stato di funzionamento del circuito, ovvero ad assistere durante la programmazione e la lettura dei tag nel normale funzionamento da serratura elettronica.

REALIZZAZIONE PRATICA

Bene, ciò detto passiamo agli aspetti costruttivi del progetto, partendo dal circuito stampato che abbiamo disegnato per accogliere i due moduli (Arduino Nano e il VMA405N) oltre ai pochi componenti discreti occorrenti: la base è a doppia faccia e potete ottenerla per fotoincisione a partire dalle tracce lato rame disponibili per il download sul nostro sito web elettronica.in.it, oppure richiedendolo tramite il servizio di realizzazione PCB offerto dalla Futura Elettronica e accessibile on-line all'indirizzo futuraelettronica.it/prodotto/servizio-realizzazione-pcb. Una volta ottenuto lo stampato, potete disporvi i componenti occorrenti a partire dalle resistenze e dal diodo al silicio, passando poi al pulsante miniatura P1, ai LED e al transistor, ai condensatori e al ponte a diodi; per l'orientamento dei componenti polarizzati, ossia diodi, transistor e condensatori elettrolitici, ponte a diodi, seguite l'indicazione fornita dal piano di montaggio che trovate in queste pagine.

Procedete poi inserendo e saldando nelle rispettive piazzole le due morsettiere, una bipolare e l'altra tripolare, rispettivamente per l'alimentazione (PWR) e i tre contatti dello scambio del relé. I morsetti sono rivolti verso l'interno del PCB. Si deve quindi praticare un foro sulla base del contenitore per far passare il cavo.

Per ospitare la scheda Arduino Nano e il modulo RFID, i quali andranno entrambi provvisti di pin-strip a passo 2,54 mm, inserite e saldate degli strip femmina nelle piazzole ad essi riservate: ne occorrono uno da 8 poli per il modulo VMA405N e due da 15 poli per la scheda Arduino Nano. Fatto ciò introducete le due board rispettando l'orientamento indicato nel piano di montaggio e, seppure appaia scontato, con il lato componenti rivolto verso l'alto.

Il modulo RFID deve essere fissato mediante quattro torrette distanziali a vite 3MA lunghe 15mm, in modo da farci stare il relé e vari componenti al di sotto, ma anche per avvicinarlo alla superficie del contenitore il più possibile.

Un esempio di realizzazione può essere fornito dalla **Fig. 2**, che propone l'assemblaggio del nostro prototipo sulla base di un contenitore in plastica di tipo Teko COFFER 1. A questo punto il vostro circuito e non vi resta che caricare in Arduino Nano lo sketch che permetterà al circuito da funzionare come elettroserratura; allo scopo dovete, senza alimentare il circuito tramite la morsettiera PWM, connettere la presa miniUSB di Arduino Nano a un Personal Computer con installato l'IDE Arduino (versione almeno 1.6) attraverso un cavo USB-A/microUSB.

Fatto ciò, avviato l'Arduino IDE bisogna verificare dal comando di menù Strumenti>Scheda che sia presente nel sotto-menù la board Arduino Nano, diversamente andrà installata; comunque di norma questa scheda rientra tra quelle installate per default all'installazione dell'IDE, pertanto non dovrebbero esserci problemi.



Fig. 2
Il prototipo
fissato alla
base del
contenitore.

Aprirete lo sketch (di cui nel **Listato 1** trovate la parte riguardante il setup, mentre nel **Listato 2** potete vedere la parte riguardante la verifica di presenza e la gestione del tag RFID) e caricatelo

Listato 1

```
//***** S E T U P *****
void setup()
{
  Serial.begin(9600); // Initialize serial communications with the PC
  while (!Serial);    // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
  SPI.begin();        // Init SPI bus

  pinMode (LED_R, OUTPUT);
  pinMode (LED_V, OUTPUT);
  pinMode (RELAY, OUTPUT);
  pinMode (PUSH, INPUT_PULLUP);
  pinMode (SECURITY_MASTER, INPUT_PULLUP);
  digitalWrite(LED_R, LOW);
  digitalWrite(LED_V, LOW);
  digitalWrite(RELAY, LOW);
  for (int i=0; i<5; i++)
  {
    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_V, HIGH);
    delay(200);
    digitalWrite(LED_R, LOW);
    digitalWrite(LED_V, LOW);
    delay(200);
  }
  mfrc522.PCD_Init();
  Serial.println("System Startup V1.0");

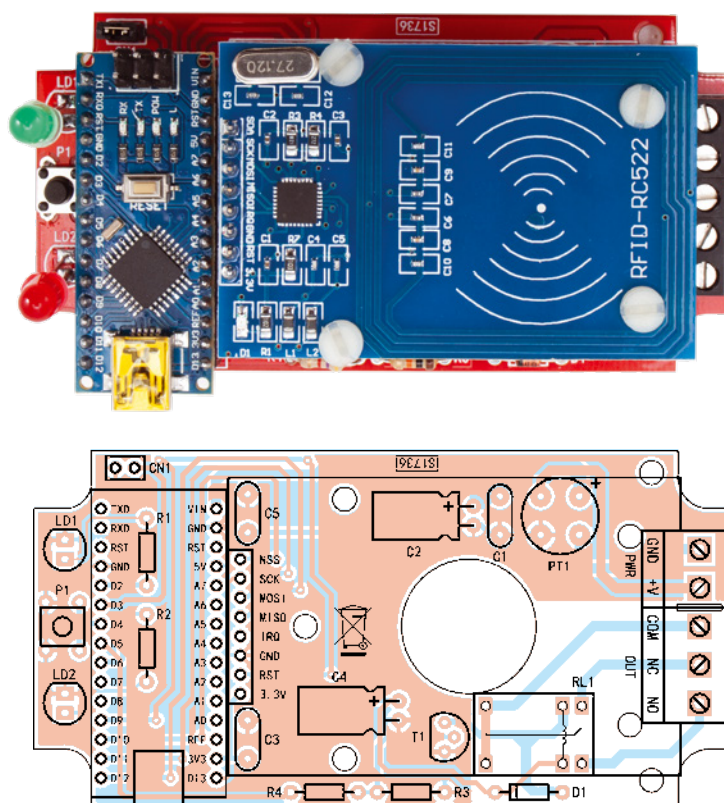
  //Cancella tutta la memoria
  if (digitalRead(PUSH) == LOW)
  {
    Serial.println("Modo: Reset Memoria");
    digitalWrite(LED_R, HIGH);
    digitalWrite(LED_V, HIGH);
    Mem_Canc();
    while (digitalRead(PUSH) == LOW) delay(100);
    delay(2000);
    digitalWrite(LED_R, LOW);
    digitalWrite(LED_V, LOW);
  }
}
```

Elenco Componenti:

R1, R2: 470 ohm
 R3: 10 kohm
 R4: 4.7 kohm
 C1, C3, C5: 100 nF ceramico
 C2, C4: 100 µF 25 VL elettrolitico
 T1: BC547
 LD1: LED 5 mm verde
 LD2: LED 5 mm rosso
 P1: Microswitch
 RL1: Relè 5V
 D1: 1N4148
 PT1: Ponte diodi W02M
 U1: Arduino Nano (MF1736)
 U2: Modulo NFC 13.56 MHz

Varie:

- Morsetto 2 poli passo 5.08 mm
- Morsetto 3 poli passo 5.08 mm
- Strip femmina 8 vie passo 2.54 mm
- Strip femmina 15 vie passo 2.54 mm (2 pz.)
- Strip maschio 2 vie
- Distanziale F/F 15 mm (4 pz.)
- Vite nylon M3 10 mm (8 pz.)
- Jumper
- Circuito Stampato S1736 (93x47 mm)



in Arduino Nano, quindi una volta ottenuto, nella parte bassa della finestra dell'IDE, il messaggio con la conferma dell'avvenuto caricamento con successo, sconnettete il cavo USB; ora il vostro circuito è operativo.

Prima di alimentarlo impostate il jumper MODE (CN1) come preferite, quindi procuratevi un alimentatore da rete con uscita a 12Vcc, meglio se stabilizzata, capace di erogare almeno 400 mA, poi collegatelo alla morsettiera +/- PWR. Ora siete pronti ad acquisire e abbinare i tag che utilizzerete per intervenire sul relè del circuito.

UTILIZZO DELLA SERRATURA RFID

Bene, una volta che avete montato, verificato e programmato il vostro circuito, siete pronti a utilizzarlo, fermo restando che fin quando non abbinerete ad esso almeno un tag, non vi servirà a molto; vediamo dunque qui di seguito come si abbinano i tag RFID memorizzandone i codici corrispondenti nella EEPROM del microcontrollore della scheda Arduino Nano, come si cancella la memoria e in che modo si gestisce la modalità di funzionamento, partendo da quest'ultima.

Modalità di sicurezza

La procedura di memorizzazione e cancellazione del tag per l'accesso, può avvenire previa autorizzazione di un utente Master (CN1 chiuso), oppure senza alcuna autorizzazione (CN1 aperto). Nel caso serva l'autorizzazione del Master, servirà il tag Master per poter cancellare e/o memorizzare un tag.

Nel caso invece si lasci CN1 aperto, la memorizzazione e la cancellazione potranno avvenire senza alcuna tessera di consenso.

Memorizzazione dei Tag

La memorizzazione del tag Master avviene unicamente nella prima posizione di memoria e non si potranno memorizzare altri Master. Qualora si smarrisca la tessera (tag) Master, dovrà essere cancellata tutta la memoria, per poi memorizzare nuovamente tutti i tag. Quando il jumper CN1 è chiuso, se la memoria è stata cancellata, oppure si sta memorizzando il primo tag, quest'ultimo sarà riconosciuto come Master. In caso di successive memorizzazioni di ulteriori tag, sarà necessario disporre del Tag Master

Listato 2

```
//Se card letta, allora la elaboro
if (mfr522.PICC_IsNewCardPresent() && mfr522.PICC_ReadCardSerial())
{
    Serial.print("\nID Tag:");
    dump_byte_array(mfr522.uid.uidByte, mfr522.uid.size);
    mfr522.PICC_HaltA();
    mfr522.PCD_StopCrypto1();

    //Utilizzo serratura
    if (MemCanc == 0)
    {

        //Card trovata
        if (Ctrl_Card(mfr522.uid.uidByte) != 255)
        {
            Serial.println("Accesso: Consentito");
            digitalWrite(RELAY, HIGH);
            digitalWrite(LED_V, HIGH);
            delay(RELAY_Time);
            digitalWrite(RELAY, LOW);
            digitalWrite(LED_V, LOW);
        }

        //Card non trovata
        else
        {
            Serial.println("Accesso: Negato");
            digitalWrite(LED_R, HIGH);
            delay(1000);
            digitalWrite(LED_R, LOW);
        }
    }
}
```

per memorizzare gli altri tag.

Per abbinare un tag bisogna premere e tenere premuto P1 fino a quando si illuminerà il solo LED verde (circa 3 secondi), quindi rilasciare il pulsante.

- **Memoria Vuota.** A questo punto se è la prima memorizzazione, il LED rimarrà fisso in attesa del passaggio di tessere. Quando una nuova tessere verrà passata, il LED verde di spegnerà per un secondo per poi riaccendersi in attesa della tessera successiva ad indicare l'avvenuta memorizzazione. Ovviamente la prima tessera sarà la Master.
- **Master Memorizzato.** Il LED verde e il LED rosso lampeggeranno alternativamente ad indicare che il lettore è in attesa della tessera Master. Quando la tessera Master verrà passata sul lettore, il LED verde rimarrà acceso fisso, in attesa delle tessere da memorizzare. Il LED verde si spegnerà per un secondo per poi riaccendersi in attesa della tessera successiva ad indicare l'avvenuta memorizzazione.

Se in jumper CN1 è aperto, bisogna premere e tenere premuto P1 fino a quando si illuminerà il

solo LED verde (circa 3 secondi), quindi rilasciare il pulsante.

Il LED rimarrà fisso in attesa del passaggio di tessere. Quando una nuova tessere verrà passata, il LED verde di spegnerà per un secondo per poi riaccendersi in attesa della tessera successiva ad indicare l'avvenuta memorizzazione.

Se il tag è già presente in memoria, il LED verde non si spegnerà ad indicare che la memorizzazione non è avvenuta. Nel caso in cui si sia raggiunto il limite di memoria, lampeggerà il led rosso e la procedura di memorizzazione verrà abbandonata.

Cancellazione Tag

In questo caso viene eliminato dalla memoria il singolo Tag che può essere cancellato solo se si ha il Tag a disposizione, in caso contrario dovrà essere ripristinata tutta la memoria.

Qualora venga cancellato il Tag Master, alla successiva memorizzazione, il primo Tag memorizzato, sarà il Master.

Se CN1 è chiuso, premere e tenere premuto P1 fino a quando si illuminerà il solo LED rosso (circa 6 secondi), quindi rilasciare il pulsante.



Il LED verde e il LED rosso lampeggeranno alternati ad indicare che il lettore è in attesa della tessera Master.

Quando la tessera Master verrà passata sul lettore, il LED rosso rimarrà acceso fisso, in attesa delle tessere da cancellare. Il LED rosso si spegnerà per un secondo per poi riaccendersi in attesa della tessera successiva ad indicare l'avvenuta cancellazione.

Invece se CN1 è aperto, bisogna premere e tenere premuto P1 fino a quando si illuminerà il solo LED rosso (circa 6 secondi), quindi rilasciare il pulsante. Il LED rosso rimarrà acceso a luce fissa, in attesa delle tessere da cancellare.

Il LED rosso si spegnerà per un secondo per poi riaccendersi in attesa della tessera successiva ad indicare l'avvenuta cancellazione.

Per uscire dalla procedura di Memorizzazione o Cancellazione di un Tag, premere il pulsante P1 per un breve istante (il LED rosso e verde si illumineranno insieme), quindi rilasciare il pulsante.

A conferma, i LED si spegneranno.

Le parti di codice riguardanti le fasi di memorizzazione e di cancellazione di un tag (scheda) RFID sono meglio descritte nelle porzioni riportate nel **Listato 3**, adeguatamente commentate entrambe.

Cancellazione totale memoria

La cancellazione totale può avvenire esclusivamente durante la fase di avvio.

Pertanto, premere e tenere premuto il pulsante di reset bianco posto sopra Arduino e il pulsante nero (P1) presente sul circuito base. Successivamente, rilasciare solo il pulsante bianco e attendere che i LED verde e rosso si accendano fissi, quindi rilasciare anche P1.

La cancellazione della memoria dei tag RFID è dettagliata nella porzione di codice proposta nel **Listato 4**, contenente anche la parte di verifica delle locazioni libere in cui poter memorizzare codici di nuove card.

Listato 3

```
//Memorizzazione Scheda
if ((MemCanc == 1) && (Attesa_Master == false))
{
    Address = Ctrl_Card(mfrc522.uid.uidByte);
    //Se la card non è già stata trovata in memoria, si può memorizzare
    if (Address == 255)
    {
        Address = Mem_Libera_LOC();
        EEPROM.write(Address, 0);
        EEPROM.write(Address+1, mfrc522.uid.uidByte[0]);
        EEPROM.write(Address+2, mfrc522.uid.uidByte[1]);
        EEPROM.write(Address+3, mfrc522.uid.uidByte[2]);
        EEPROM.write(Address+4, mfrc522.uid.uidByte[3]);
        Serial.println("Card Memorizzata");
        digitalWrite(LED_V, LOW);
        delay(500);
        Mem_Libera();
    }
}

//Cancellazione Scheda
if ((MemCanc == 2) && (Attesa_Master == false))
{
    Address = Ctrl_Card(mfrc522.uid.uidByte);
    if (Address != 255)
    {
        EEPROM.write(Address, 255);
        Serial.println("Card Cancellata");
        digitalWrite(LED_R, LOW);
        delay(1000);
        digitalWrite(LED_R, HIGH);
    }
}
```

Listato 4

```

//***** M E M _ L I B E R A *****
//Verifica la presenza di memorie libere in memoria
//Se presenti lcoazioni libere, accende il LED verde, in alternativa fa lampeggiare il rosso ed esce
int Mem_Libera()
{
  int Loc_Libere = 0;
  for (int i=0; i<Max_Card; i++)
  {
    if (EEPROM.read(i * Byte_Card) == 255) Loc_Libere += 1;
  }
  if (Loc_Libere > 0)
  {
    digitalWrite(LED_V, HIGH);
    digitalWrite(LED_R, LOW);
  }

  else //Memoria piena quindi si esce dalla memorizzazione
  {
    Attesa_Master = false;
    digitalWrite(LED_V, LOW);
    for (int i=0; i<5; i++)
    {
      digitalWrite(LED_R, HIGH);
      delay(200);
      digitalWrite(LED_R, LOW);
      delay(200);
    }
    MemCanc = 0;
  }
}

//***** M E M _ C A N C *****
//Cancellazione totale della memoria
int Mem_Canc()
{
  for (int i=0; i<Max_Card; i++)
  {
    Address = i * Byte_Card;
    EEPROM.write(Address,255);
  }
}

```

Funzionamento della serratura

Quando il circuito viene alimentato, durante il normale funzionamento, fornendo alimentazione a 12V, il LED verde e rosso lampeggeranno 5 volte ad indicare il corretto avvio. Terminata la fase di avvio, la serratura, si porrà in standby, in attesa di un tag sul lettore. Quando il tag verrà avvicinato, se il suo ID è stato abilitato (tag salvato in memoria ovvero abbinato al sistema) il relé si attiverà per 1 secondo insieme al LED verde, per poi disattivarsi. Qualora l'ID non sia stata trovato in memoria, si illuminerà il solo LED rosso e il relé non commuterà.

CONCLUSIONI

Bene, termina qui la descrizione del progetto di serratura elettronica basata su RFID, che nel caso specifico è stata affidata a un modulo lettore/

programmatore di tag operanti alla frequenza standard di 13,56 MHz e ad una scheda Arduino che, interfacciata ad esso, gestisce un relé. 



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
FT1736K	Chiave RFID 13,56 MHz completa	49,00
1356RFIDCARD	Tessera RFID 13,56 MHz	0,70
1356RFIDTAG	TAG RFID 13,56 MHz	0,80

IL MATERIALE VA RICHIESTO A:

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



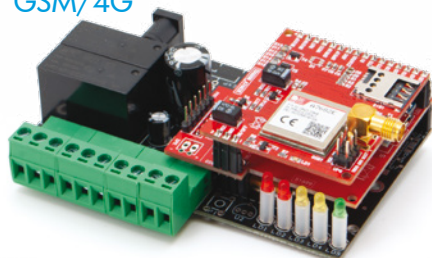
Gestisci e controlla la tua casa con i TELECONTROLLI GSM

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. Configurazione via SMS/PC con predisposizione per gruppo di continuità.

4G
LTE



COMBINATORE TELEFONICO GSM/4G

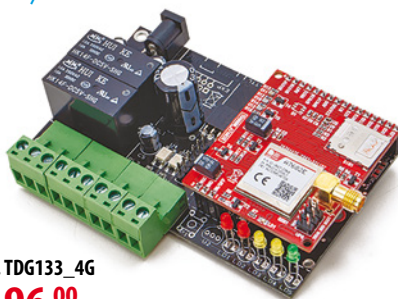


cod. TDG135MP3_4G
€125,00

- 2 Ingressi e 2 Uscite
- Menu vocale in Italiano
- Messaggi di allarme vocali
- Invio segnalazioni di allarme ingressi

Disponibile anche in versione 2G cod. TDG135MP3 • Euro 99,00

TELECONTROLLO BICANALE GSM/4G

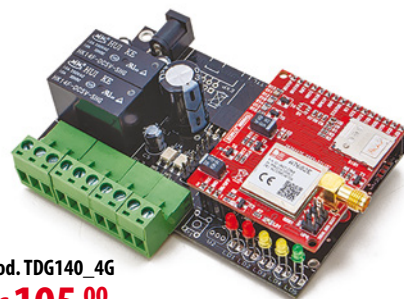


cod. TDG133_4G
€96,00

- 2 Ingressi e 2 Uscite
- Funzione Apricancello
- Invio segnalazioni di allarme ingressi

Disponibile anche in versione 2G cod. TDG133 • Euro 79,00

TELECONTROLLO GSM/4G CON COMANDI DTMF

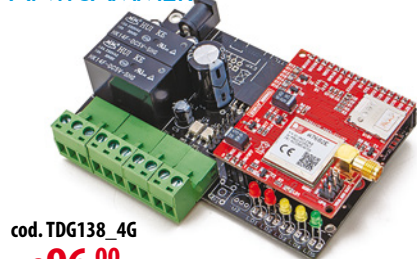


cod. TDG140_4G
€105,00

- 2 Ingressi e 2 Uscite
- Menu gestibile via DTMF
- Invio segnalazioni di allarme ingressi

Disponibile anche in versione 2G cod. TDG140 • Euro 88,00

TELEALLARME GSM/4G CON ANTI JAMMER

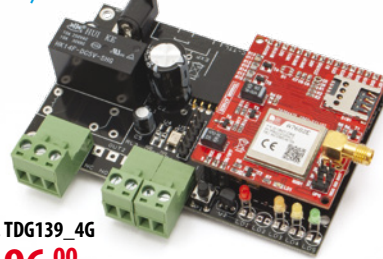


cod. TDG138_4G
€96,00

- 2 Ingressi e 2 Uscite
- Funzione anti Jammer o Telecontrollo
- Invio segnalazioni di allarme ingressi e Jammer

Disponibile anche in versione 2G cod. TDG138 • Euro 79,00

TERMOSTATO CON CONTROLLO GSM/4G



cod. TDG139_4G
€96,00

- Controllo riscaldamento
- Manuale/Automatiko/Asservito Termostato
- Allarmi temperatura configurabili
- Invio segnalazioni di allarme temperatura e blocco caldaia

Disponibile anche in versione 2G cod. TDG139 • Euro 79,00

APRICANCELLO GSM/4G

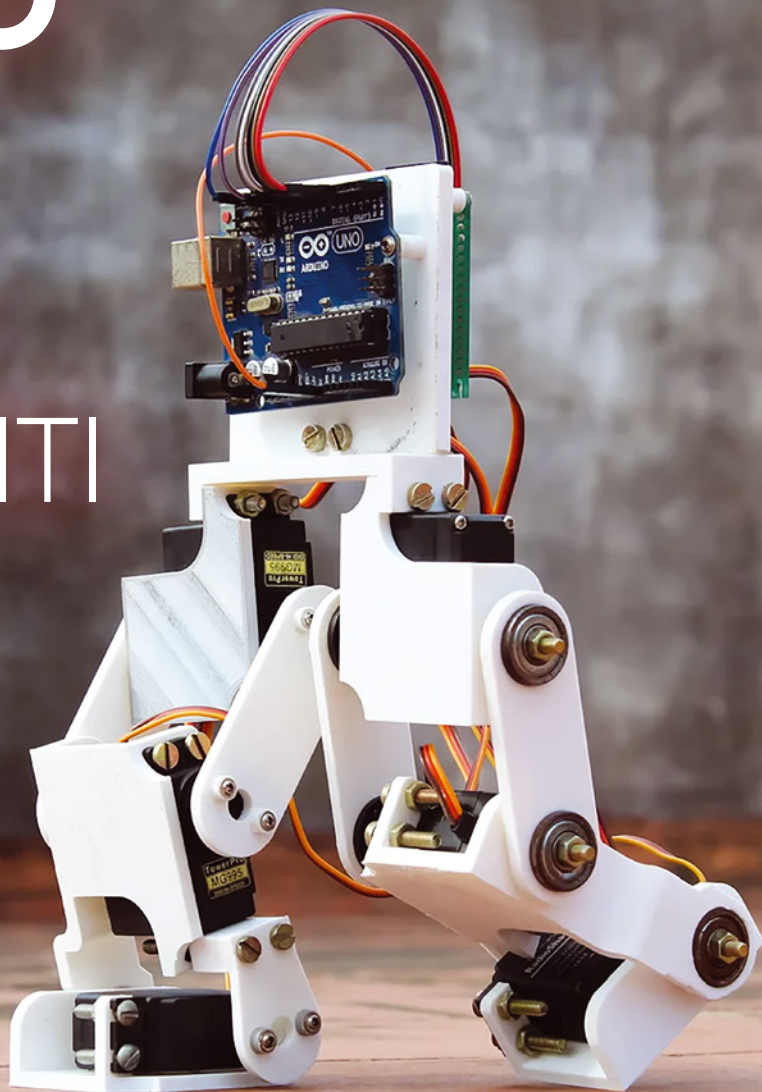
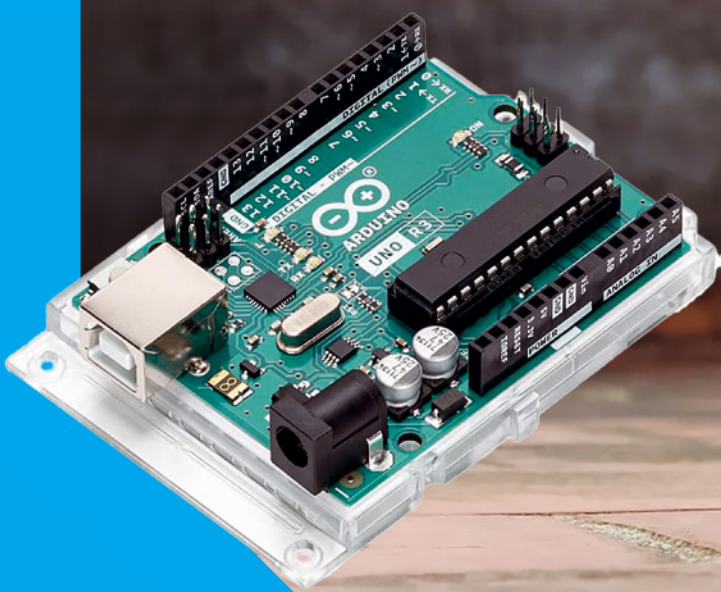


cod. TDG134_4G
€92,00

- 1 Ingresso
- Funzione Apricancello

Disponibile anche in versione 2G cod. TDG134 • Euro 75,00

ARDUINO E GLI AUTOMI A STATI FINITI



di **PAOLO GUIDI**

Scopriamo le macchine a stati finiti, utili in automazione e hobbistica per un design chiaro e semplice, fondamentale in elettronica e automazione industriale.

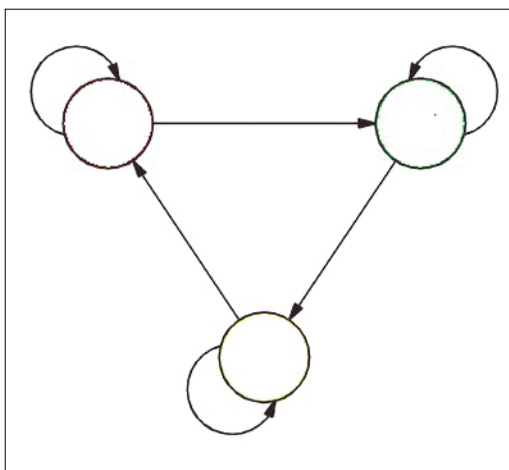
U

na macchina a stati finiti (FSM dall'inglese *finite state machine*) o automa a stati finiti (ASF o FSA dall'inglese *finite state automaton*) è un modello che è in grado di descrivere efficacemente il comportamento di un sistema che presenta un numero *finito* di stati e che in ogni istante si può trovare in uno solo

di essi definito con il termine *stato corrente*; il sistema può passare da uno stato all'altro solo se si verificano determinate condizioni e tale passaggio viene definito con il termine *transizione*.

Un modello di questo tipo è molto diffuso in ambito scientifico in quanto il modo di procedere risulta sempre caratterizzato da semplicità e chiarezza; è applicabile in particolar modo anche nel settore dell'automazione industriale.

→ **Fig. 1**
La rappresentazione di un grafo orientato.



L'impiego di una progettazione basata sul concetto di macchina a stati finiti consente infatti di realizzare più facilmente apparati elettronici e sistemi. Il metodo, una volta acquisito, può anche essere utilizzato dai maker (hobbisti tecnologici) che, grazie all'avvento di schede come Arduino, coprono un settore che è sempre più praticato anche da coloro che non posseggono conoscenze approfondite in ambito elettronico/informatico o che, addirittura, partono da zero. Anche chi ha un minimo di conoscenze di base spesso non si accorge che le macchine a stati finiti sono costantemente presenti nella vita quotidiana; si citano come esempi l'ascensore, la lavatrice, il distributore automatico di bibite o di merendine, il semaforo, il cancello automatico. Visto come macchina a stati finiti l'ascensore può ad esempio essere considerato

come un automa che presenta un numero di stati pari a quello dei piani; la transizione da uno stato all'altro viene definita dai pulsanti di chiamata posti ai piani e dai pulsanti che si trovano all'interno della cabina.

UNO STRUMENTO GRAFICO FONDAMENTALE: IL DIAGRAMMA DEGLI STATI

Una macchina a stati finiti si può definire utilizzando un *diagramma degli stati* che può essere descritto graficamente attraverso un *grafo orientato*.

Un grafo orientato (**Fig. 1**) è una struttura che viene nella maggior parte dei casi rappresentata da un numero finito di cerchi (*nodi*) e da un numero finito di archi provvisti di freccia (*archi orientati*) che collegano ciascun nodo agli altri nodi; gli archi dei grafi orientati hanno una direzione.

I nodi rappresentano gli stati a cui vengono associate le azioni da implementare all'interno degli stati stessi (ad esempio l'accensione di una lampada o l'avviamento di un motore); gli archi (talvolta sostituiti con tratti rettilinei anch'essi orientati) definiscono le condizioni per le quali si verificano le transizioni di stato causate da eventi esterni (ad esempio la pressione di un pulsante).

PROGETTO: METODOLOGIE

Dopo questa breve introduzione si prosegue con la descrizione del procedimento che consente la realizzazione di una macchina a stati (lo schema sintetico viene riportato in **Fig. 2**).

È comunque bene precisare subito che il passaggio che dall'analisi del problema porta alla definizione del grafo orientato è praticamente indipendente dal tipo di ambiente di sviluppo che viene utilizzato e assume pertanto validità generale.

Dall'analisi del problema si ricavano il numero di stati dell'automa, le condizioni necessarie per passare da uno stato ad un altro e, di conseguenza, il grafo orientato.

Il passo successivo, quello che consente di creare un legame tra diagramma degli stati e codice, dipende in parte dall'ambiente di sviluppo scelto; si procede di seguito facendo riferimento ad una scheda Arduino, quella più utilizzata e di più semplice impiego. Per la scrittura del codice si può utilizzare un procedimento standard.

In fase di programmazione si tratta di tradurre le informazioni fornite dal grafo orientato in una struttura a selezione multipla (*switch...case*) avente un numero di *case* pari al numero di stati. Ad ogni *case* si fa corrispondere un sottoprogramma ed è all'interno di questo che, utilizzando

analisi del problema



grafo orientato



scrittura del codice

→ **Fig. 2**
Schema del procedimento per la realizzazione di una macchina a stati.

in modo opportuno strutture di selezione e/o di iterazione (*if*, *while*, *for*), si definiscono le condizioni per il passaggio da uno stato ad un altro nonché l'esecuzione delle azioni che, stato per stato, devono essere implementate.

I *case* che si devono di volta in volta eseguire dipendono dal valore che si assegna alla variabile su cui opera l'istruzione (definita anche con il termine *selettore*).

Il modo di procedere stilisticamente e intuitivamente più valido è quello ora descritto.

Una possibile alternativa è quella di non utilizzare i sottoprogrammi ma di inserire le istruzioni che consentono di definire i passaggi di stato e le azioni da implementare all'interno dei *case*.

Si procede di seguito con alcuni esempi facilmente realizzabili che consentano di comprendere il procedimento poi applicabile a situazioni più complesse.

UN PRIMO ESEMPIO: IL DEVIATORE DEGLI IMPIANTI ELETTRICI CIVILI

Il deviatore è un dispositivo molto comune negli impianti elettrici civili tradizionali che consente di accendere un punto luce da due punti distanti l'uno dall'altro; lo schema elettrico dell'impianto (denominato anche deviato) viene schematizzato in **Fig. 3**. La rappresentazione dell'automa che ne definisce il funzionamento è molto semplice in quanto l'automa stesso presenta due stati che sono le possibili condizioni OFF e ON della lampada L (denominata LED) definiti stato 0 e stato 1; le condizioni che si possono verificare per passare da OFF a ON e viceversa dipendono da come vengono azionati i due pulsanti (denominati P1 e P2) che sostituiscono i deviatori D1 e D2.

Il funzionamento può essere descritto molto semplicemente (grafo orientato di **Fig. 4**) in quanto è sufficiente premere uno qualsiasi dei due pulsanti per passare da uno stato all'altro.

Lo schema elettrico riportato in **Fig. 5** si riferisce alla realizzazione pratica dell'automa che utilizza la scheda Arduino UNO.

Il codice rappresentativo dell'automa di seguito riportato è quello di Arduino; il pulsante P1 è stato collegato al pin 10, il pulsante P2 al pin 12, il LED infine al pin 8.

La presenza di due stati porta immediatamente a definire una struttura *switch...case* con due *case*; i sottoprogrammi che ne conseguono definiscono di pertanto lo stato del LED e le condizioni relative ai passaggi di stato.

Nella soluzione alternativa di seguito proposta le

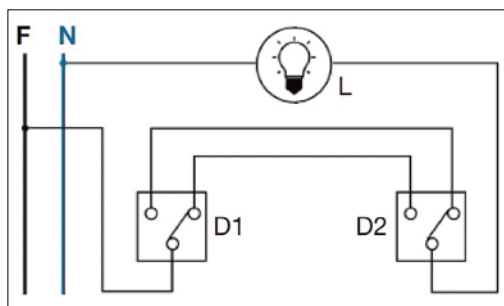


Fig. 3
Schema di una deviato.

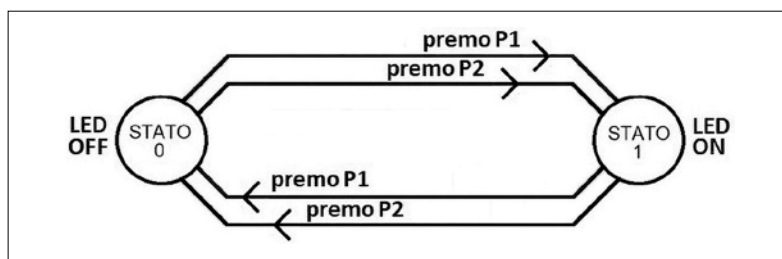


Fig. 4
Grafo orientato della deviato.

istruzioni che consentono di definire i passaggi di stato vengono inserite all'interno dei *case*. Come possibile variante relativa a questo esempio potrebbe essere una ulteriore applicazione utilizzata negli impianti civili tradizionali ovvero l'accensione di un punto luce da tre punti (l'impianto viene comunemente denominato interrotta).

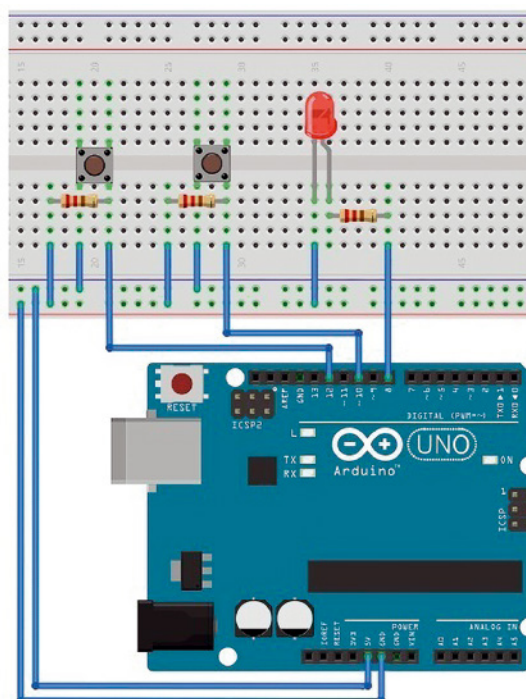


Fig. 5
Una deviato realizzata con Arduino.

Listato 1

```
int stato=0;          // variabile selettore
void setup(){
  pinMode(8,OUTPUT); // pin 8 uscita LED
  pinMode(10,INPUT); // pin 10 pulsante 1
  pinMode(12,INPUT); // pin 12 pulsante 2
}
void loop(){
  switch(stato){
    case 0: stato0(); break;
    case 1: stato1(); break;
  }
}
void stato0(){
  digitalWrite(8,LOW); // il LED è spento
  if(digitalRead(10)==HIGH){ // leggo pulsante 1
    stato=1;delay(1000);}
  if(digitalRead(12)==HIGH){ // leggo pulsante 2
    stato=1;delay(1000);}
}
void stato1(){
  digitalWrite(8,HIGH); // il LED 1 è acceso
  if(digitalRead(10)==HIGH){ // leggo pulsante 1
    stato=0;delay(1000);}
  if(digitalRead(12)==HIGH){ // leggo pulsante 2
    stato=0;delay(1000);}
}
```

Listato 2

```
int stato=0;          // variabile selettore
void setup(){
  pinMode(8,OUTPUT); // pin 8 uscita LED
  pinMode(10,INPUT); // pin 10 pulsante 1
  pinMode(12,INPUT); // pin 12 pulsante 2
}
void loop(){
  switch(stato){
    case 0:
      digitalWrite(8,LOW); // il LED è spento
      if(digitalRead(10)==HIGH){ // leggo pulsante 1
        stato=1;delay(1000);}
      if(digitalRead(12)==HIGH){ // leggo pulsante 2
        stato=1;delay(1000);};break;
    case 1:
      digitalWrite(8,HIGH); // il LED è acceso
      if(digitalRead(10)==HIGH){ // leggo pulsante 1
        stato=0;delay(1000);}
      if(digitalRead(12)==HIGH){ // leggo pulsante 2
        stato=0;delay(1000);};break;
  }
}
```

UN SECONDO ESEMPIO: ACCENSIONE IN SEQUENZA DI LED

Si considerino due LED (LED 1 e LED 2) che si accendono e spengono in sequenza a seguito della pressione di un pulsante.

Si definiscono quattro possibili stati:

- stato 0_è lo stato iniziale in cui i due LED sono spenti premendo il pulsante si passa nello stato 1
- stato 1_si accende il LED 1 premendo per la seconda volta il pulsante si passa nello stato 2
- stato 2_il LED 1 si spegne, dopo 3 s si accen-

dono entrambi premendo per la terza volta il pulsante si passa nello stato 3

- stato 3_si spengono entrambi i LED, dopo 3 s si accende il LED 2 premendo per la quarta volta il pulsante si ritorna nello stato 0

Dalla descrizione del problema si passa facilmente al grafo orientato riportato in **Fig. 6**.

Lo schema elettrico riportato in **Fig. 7** si riferisce alla realizzazione pratica dell'automa che utilizza la scheda Arduino UNO.

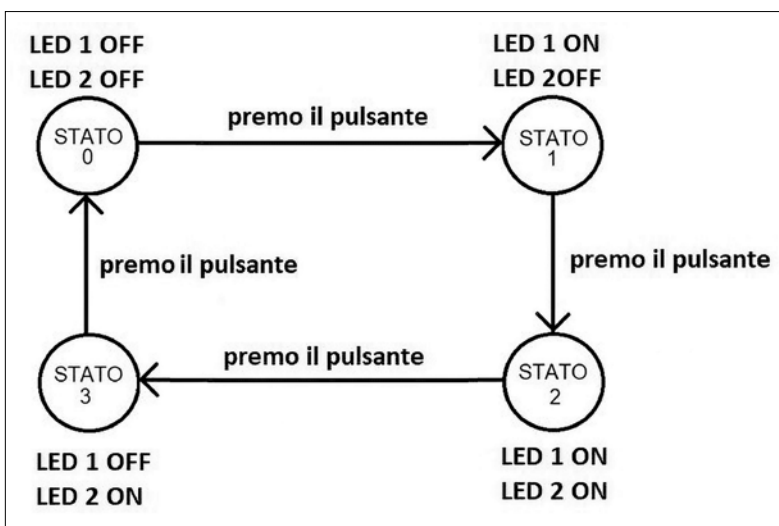
Il pulsante è stato collegato al pin 12, il LED 1 al pin 8, il LED 2 infine al pin 10. Il codice rappresentativo dell'automa viene di seguito riportato. La presenza di quattro stati porta immediatamente a definire una struttura *switch...case* con quattro *case*; i sottoprogrammi che ne conseguono definiscono di conseguenza lo stato dei LED e le condizioni relative ai passaggi di stato.

UN TERZO ESEMPIO: VERSO L'AUTOMAZIONE INDUSTRIALE

In questo terzo esempio vengono mostrate in modo molto semplificato alcune azioni che sono tipiche dell'automazione industriale.

In **Fig. 8** viene riprodotto un nastro trasportatore il cui movimento dipende dallo stato assunto da un pulsante (per l'avvio del nastro), da due finecorsa (definiti di andata e di ritorno posti agli estremi del nastro stesso) e da una fotocellula posta in posizione intermedia.

Fig. 6
Grafo orientato per l'accensione in sequenza dei led.



Il funzionamento dell'impianto può essere sintetizzato nel modo seguente definendo quattro possibili stati:

- stato 0_è lo stato iniziale in cui il motore è fermo
- premendo il pulsante PA si passa nello stato 1
- stato 1_ il nastro è avviato nella modalità marcia avanti MA

Dallo stato 1 si possono seguire due percorsi ovvero passare allo stato 2 (quando viene rilevato lo stato ON della cellula fotoelettrica) o direttamente allo stato 3 (quando viene rilevato lo stato ON del finecorsa FCA):

- stato 2_ il motore si ferma per 60 s e riprende il movimento sempre in modalità MA
- quando viene rilevato lo stato ON del finecorsa FCR si passa nello stato 3
- stato 3_ il motore si ferma per 60 s e riprende il movimento in modalità MI; quando viene rilevato lo stato ON del finecorsa FCR si ritorna nello stato 0

Dalla descrizione del problema si passa facilmente al grafo orientato riportato in **Fig. 9**.

Per verificare la correttezza di quanto affermato vengono utilizzati due LED (che simulano rispettivamente la marcia avanti e la marcia indietro del motore), un pulsante (che simula il pulsante di avvio) e tre pulsanti (che simulano il comportamento dei tre sensori).

Lo schema elettrico riportato in **Fig. 10** si riferisce alla realizzazione pratica dell'automa che utilizza la scheda Arduino UNO.

Il codice rappresentativo dell'automa viene di seguito riportato.

CONCLUSIONI

Abbiamo visto come le macchine a stati finiti (FSM) siano fondamentali per la modellazione di sistemi con comportamenti deterministici, giocando un ruolo cruciale nell'ingegneria del controllo e dell'automazione.

Le FSM offrono un quadro metodologico solido per delineare sistemi complessi attraverso stati ben definiti e transizioni precise, facilitando la progettazione di circuiti elettronici e sistemi di automazione industriale.

Grazie alla loro struttura rigorosa e alla capacità di rappresentare chiaramente ogni possibile stato di un sistema, le FSM aiutano a prevenire errori e migliorare l'affidabilità dei processi automatizzati. Abbiamo anche scoperto l'importanza delle piattaforme come Arduino nell'implementazione pratica

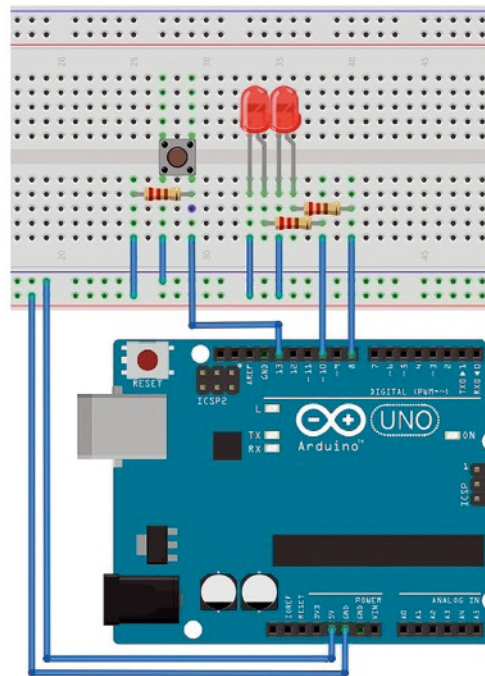


Fig. 7
Schema di cablaggio per l'accensione in sequenza dei LED.

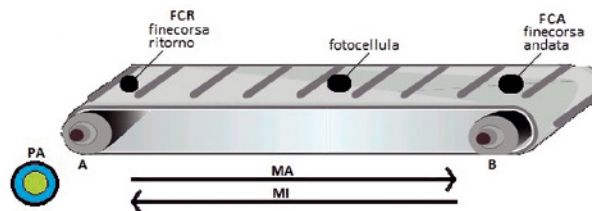


Fig. 8
Simulazione di un nastro trasportatore.

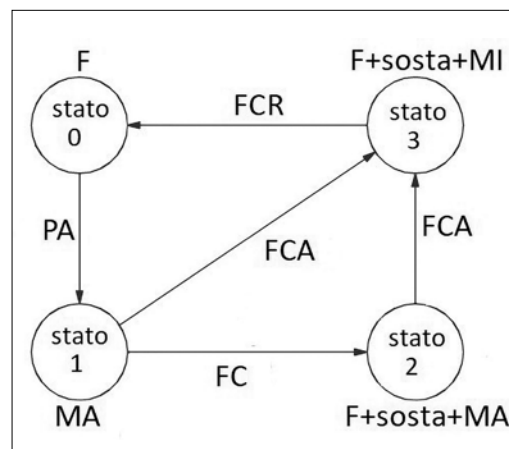


Fig. 9
Grafo orientato del funzionamento di un nastro trasportatore.

Listato 3

```
int stato=0; // viene inizializzata la variabile selettore
void setup(){
  pinMode(8,OUTPUT); // definizione pin 8 come uscita LED 1
  pinMode(10,OUTPUT); // definizione pin 10 come uscita LED 2
  pinMode(12,INPUT); // definizione pin 12 come ingresso pulsante
}
void loop(){
  switch(stato){
    case 0: stato0(); break;
    case 1: stato1(); break;
    case 2: stato2(); break;
    case 3: stato3(); break;
  }
  void stato0(){
    digitalWrite(8,LOW); // il LED 1 è spento
    digitalWrite(10,LOW); // il LED 2 è spento
    if(digitalRead(12)==HIGH){ // leggo lo stato del pulsante
      stato=1;delay(1000);} // e passo allo stato successivo se premuto
    }
    void stato1(){
      digitalWrite(8,HIGH); // il LED 1 è acceso
      digitalWrite(10,LOW); // il LED 2 è spento
      if(digitalRead(12)==HIGH){ // leggo lo stato del pulsante
        stato=2;delay(1000);} // e passo allo stato successivo se premuto
      }
      void stato2(){
        digitalWrite(8,HIGH); // il LED 1 è acceso
        digitalWrite(10,HIGH); // il LED 2 è acceso
        if(digitalRead(12)==HIGH){ // leggo lo stato del pulsante
          stato=3;delay(1000);} // e passo allo stato successivo se premuto
        }
        void stato3(){
          digitalWrite(8,LOW); // il LED 1 è spento
          digitalWrite(10,HIGH); // il LED 2 è acceso
          if(digitalRead(12)==HIGH){ // leggo lo stato del pulsante
            stato=0;delay(1000);} // e passo allo stato successivo se premuto
          }
        }
```

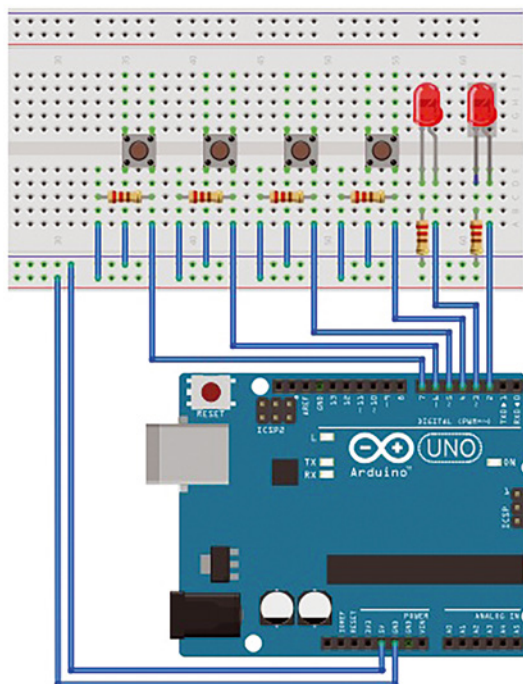


Fig. 10
Schema di cablaggio per la simulazione di un nastro trasportatore.

delle FSM, sottolineando la loro accessibilità e versatilità.

Arduino si è rivelato uno strumento prezioso non solo per gli ingegneri esperti ma anche per gli hobbisti e i principianti, democratizzando così l'accesso a tecnologie sofisticate e incentivando l'innovazione e la sperimentazione in un ampio spettro di applicazioni.

Grazie ad Arduino, chiunque può sviluppare e testare prototipi di sistemi automatizzati, rendendo la tecnologia più inclusiva e promuovendo la creatività. Attraverso vari esempi, abbiamo osservato come le FSM siano impiegate in numerosi dispositivi e sistemi quotidiani, evidenziando la loro ubiquità e importanza nell'ambito tecnologico.

Questi esempi hanno concretizzato il concetto di FSM, dimostrando come principi teorici si traducano in applicazioni pratiche essenziali per il nostro quotidiano. Dal funzionamento degli elettrodomestici agli impianti industriali complessi, le FSM sono alla base di molte tecnologie che utilizziamo ogni giorno, dimostrando la loro rilevanza e versatilità.

Listato 4

```

int stato=0; // viene inizializzata la variabile selettore
void setup(){
  pinMode(2,OUTPUT); // uscita LED marcia avanti MA
  pinMode(3,OUTPUT); // uscita LED marcia indietro MI
  pinMode(4,INPUT); // ingresso pulsante avvio PA
  pinMode(5,INPUT); // ingresso pulsante finecorsa andata FCA
  pinMode(6,INPUT); // ingresso pulsante finecorsa ritorno FCR
  pinMode(7,INPUT); // ingresso pulsante fotocellula FC
}

void loop(){
  switch(stato){
    case 0: stato0(); break;
    case 1: stato1(); break;
    case 2: stato2(); break;
    case 3: stato3(); break;}
}

void stato0(){
  digitalWrite(2,LOW); // il LED MA è spento
  digitalWrite(3,LOW); // il LED MI è spento
  if(digitalRead(7)==HIGH){ // leggo lo stato del pulsante PA
    stato=1;delay(1000);} // e passo allo stato 1 se premuto
}

void stato1(){
  digitalWrite(2,HIGH); // il LED MA è acceso
  digitalWrite(3,LOW); // il LED MI è spento
  if(digitalRead(7)==HIGH){ // leggo lo stato del pulsante FC
    stato=2;delay(1000);} // e passo allo stato 2 se premuto
  if(digitalRead(5)==HIGH){ // leggo lo stato del pulsante FCA
    stato=3;delay(1000);} // e passo allo stato 3 se premuto
}

void stato2(){
  digitalWrite(2,LOW); // il LED MA è spento
  digitalWrite(3,LOW); // il LED MI è spento
  delay(3600); // ritardo sosta
  while(digitalRead(5)==LOW){ // fino a quando il pulsante FCA non viene premuto
    digitalWrite(2,HIGH); // il LED MA è acceso
    digitalWrite(3,LOW);} // il LED MI è spento
  stato=3;delay(1000); // passo allo stato 3 se premuto
}

void stato3(){
  digitalWrite(2,LOW); // il LED MA è spento
  digitalWrite(3,LOW); // il LED MI è spento
  delay(3600); // ritardo sosta
  while(digitalRead(6)==LOW){ // fino a quando il pulsante FCR non viene premuto
    digitalWrite(2,LOW); // il LED MA è spento
    digitalWrite(3,HIGH);} // il LED MI è acceso
  stato=0;delay(1000); // passo allo stato 0 se premuto
}
    
```

Infine, abbiamo esplorato la transizione dalla teoria alla pratica delle FSM, con una guida dettagliata per la loro realizzazione utilizzando Arduino.

Questo collegamento tra la comprensione teorica e l'applicazione pratica è stato cruciale per evidenziare il potenziale delle FSM in progetti reali, aprendo nuove prospettive nell'automazione e oltre.

La capacità di tradurre la teoria in pratica non solo arricchisce la comprensione degli studenti e dei professionisti, ma stimola anche nuove idee e soluzioni innovative per le sfide tecnologiche contemporanee.



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
ARDUKITLEVEL1	Starter kit Livello base con Arduino UNO	35,90
ARDUKITLEVEL2	Starter kit Livello intermedio con Arduino UNO	45,90

IL MATERIALE VA RICHIESTO A:

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



Tutto sotto RADIO CONTROLLO

Set presa da rete radiocomandata a tre canali espandibile con telecomando



Ideale per controllare dispositivi elettrici con un carico massimo di 1100 watt. Offre la possibilità di aggiungere ulteriori trasmettitori e abbinare fino a tre prese, ciascuna associata ai tre canali operativi del trasmettitore. Particolarmente indicato per i punti difficili da raggiungere negli spazi interni, offrendo comodità e facilità d'uso. Il sistema è espandibile.

cod. WT3PRTX
€ 22,90



Telecomando
addizionale per set
WT3PRTX

cod. WT3PR
€ 14,90



Presa radiocomandata
addizionale per set
WT3PRTX

cod. WT3TX
€ 9,00

Sistema di radiocontrollo a 433 MHz

Trasmettitori a 1-2-4 canali e ricevitori a 12VDC e 220VAC operanti sulla banda dei 433 MHz. Si basano sulla tecnologia RF che fornisce un segnale stabile e a lunga distanza. I canali sono tra di loro indipendenti e consentono, oltre al controllo del carico, di eliminare qualsiasi interferenza reciproca. Bassissimo consumo in modalità stand-by. Possono essere impiegati per il controllo di luci, per l'attivazione/disattivazione di impianti d'allarme, ecc.

RICEVITORE

2 canali 220 VAC

Cod. EMY220RX2

Euro 16,00

2 canali 12 VDC

Cod. EMY12RX2

Euro 16,00

4 canali 12 VDC

Cod. EMY12RX4

Euro 19,00

TRASMETTITORE

1 canale

Cod. EMYTX1

Euro 11,00

TRASMETTITORE

2 canali

Cod. EMYTX2

Euro 12,00

TRASMETTITORE

4 canali

Cod. EMYTX4

Euro 13,00

RICEVITORE

1 canale 220 VAC

Cod. EMY220RX1

Euro 13,00

1 canale 12 VDC

Cod. EMY12RX1

Euro 13,00



Kit radiocomandi

Kit radiocomandi da esterno

2CH – 433MHz

12V – 24V – AC/DC

cod. SUPRK002

€ 30,00



Kit radiocomandi da esterno

2CH – 433MHz

220Vac



cod. SUPRK004

€ 44,00

Kit radiocomandi da interno

2CH – 433MHz

12V – 24V DC

cod. SUPRK001

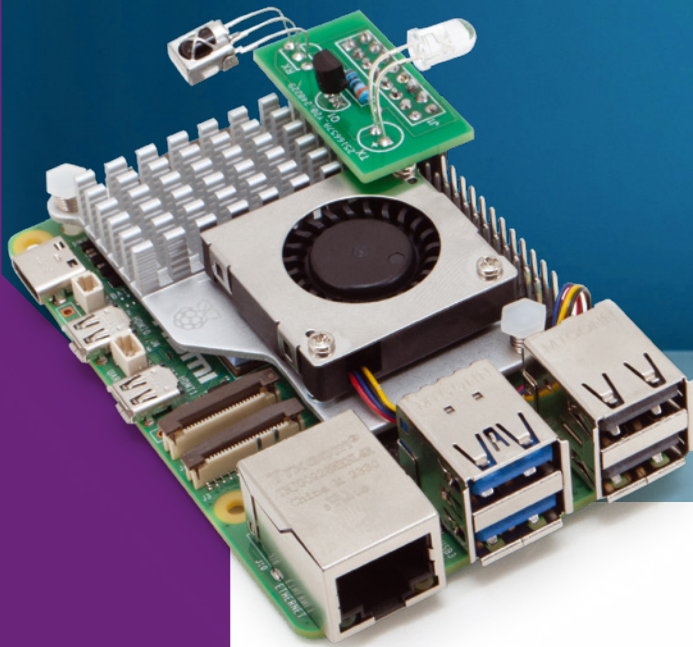
€ 24,00



Prezzi IVA inclusa.

TELECOMANDO UNIVERSALE AI MEDIAPIPE PER TV

di ALOISIO ANTONIO

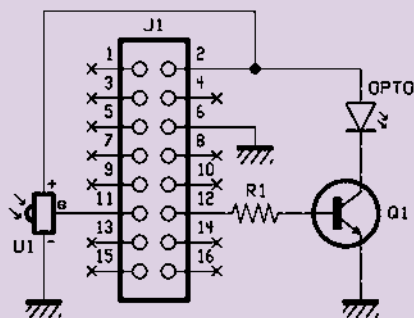


Realizziamo un dispositivo basato su Raspberry Pi capace di riconoscere gesti predefiniti della mano ed associarli a comandi ad infrarosso.

H

MI, acronimo di Human-Machine Interface, è un termine che si riferisce a qualsiasi tecnologia o dispositivo capace di far interagire gli esseri umani con le macchine. Le HMI sono fondamentali in una vasta gamma di settori, dalla produzione industriale all'elettronica di consumo, e sono essenziali per semplificare e ottimizzare le interazioni tra gli utenti e le macchine. Le HMI più usate sono:

- **Interfacce utente grafiche (GUI)**, sono le più comuni e includono schermi grafici che consentono agli utenti di interagire con le macchine attraverso icone, pulsanti e menu, tipiche ad esempio dei sistemi operativi per pc.
- **Interfacce utente vocali (VUI)**, consentono agli utenti di interagire con le macchine attraverso comandi vocali.



Questo tipo di HMI sta diventando sempre più popolare con l'avvento dell'assistenza vocale come Siri di Apple, Google Assistant e Amazon Alexa.

- **Interfacce utente tattili:** consentono agli utenti di interagire con le macchine attraverso il tocco fisico, come schermi touchscreen, tastiere e mouse.
- **Interfacce utente gestuali:** consente agli utenti di interagire con le macchine attraverso movimenti fisici, come gesti delle mani rilevati da telecamere o sensori di movimento, che sarà il focus di questo progetto.

Il campo di applicazione spazia dall'automazione industriale (utilizzate per monitorare e controllare macchine e processi industriali), all'elettronica di consumo (i touchscreen sono comuni nei dispositivi consumer come smartphone, tablet, computer e televisori). Nei sistemi domotici, le HMI consentono agli utenti di controllare luci, termostati, dispositivi di sicurezza e altri dispositivi domestici intelligenti. Sono fondamentali nei veicoli per consentire agli utenti di interagire con sistemi di navigazione, sistemi di intrattenimento e altre funzionalità. L'interazione deve seguire dettami di ergonomia e funzionalità non sempre raggiunti in tutte le HMI, basti pensare alle recenti critiche da parte di Euro NCAP, nei confronti della complessità dei veicoli che hanno spostato controlli chiave sugli schermi centrali, obbligando i conducenti a distogliere lo sguardo dalla strada, aumentando il rischio di incidenti per distrazione (fonte Quattroruote). La realizzazione di HMI più intuitive (user-friendly) è quindi uno degli obiettivi che le aziende cercano sempre più di raggiungere. Quello dell'interazione con le macchine è quindi un ambito con cui ci confrontiamo quotidianamente ed è divenuto cruciale in una vasta gamma di settori, contribuendo a migliorare l'efficienza, la sicurezza e l'esperienza complessiva degli utenti. Oggi, molti dei sistemi HMI sfruttano l'intelligenza

artificiale (AI) per il loro funzionamento.

Il progetto che vi proporremo in questo articolo, userà il riconoscimento dei gesti della mano per dare dei comandi ad un televisore. Useremo un sistema di AI per sostituire uno dei dispositivi più comuni e "rassicuranti" nella vita di tutti i giorni, il telecomando. Alla fine del progetto sfidiamo chiunque a non divertirsi accendendo il televisore con "la forza Jedi" dell'apertura del palmo della mano. L'approccio del progetto è ludico, ma vuole essere una porta di ingresso sull'uso dei sistemi di intelligenza artificiale (AI) da parte di tutti in questo caso mirata alla creazione di HMI gestuale.

SOFTWARE

L'intelligenza artificiale (AI) è un campo dell'informatica che si concentra sullo sviluppo di sistemi e tecnologie in grado di simulare i processi di intelligenza umana, cioè pensare e risolvere problemi a cui siamo abituati noi esseri umani.

Il "machine learning" (ML) è invece un sottoinsieme fondamentale della AI che studia come far sì che un sistema informatico possa imparare da **dati** passati per fare predizioni o prendere decisioni, senza essere esplicitamente programmati per farlo. Il primo passo nel ML è l'acquisizione dei dati che possono essere di vario tipo, come testo, immagini, suoni o numeri. Possono provenire da sensori, database o persino dai social media. L'importanza dei dati di qualità è fondamentale nel determinare il successo di un modello di ML.

Gli **algoritmi** di ML sono invece gli strumenti che permettono ai sistemi informatici di imparare dai dati. Esistono vari tipi di algoritmi, ognuno dei quali è adatto a compiti specifici. Ad esempio, gli algoritmi di regressione possono essere utilizzati per fare previsioni basate su dati storici, mentre gli algoritmi di classificazione sono utili per assegnare categorie ai dati.

Viene poi il processo di **addestramento** di un sistema (algoritmo + dato) di ML, cioè l'alimentazione dell'algoritmo con dati etichettati, cioè dati che hanno già delle risposte corrette associate. L'algoritmo utilizza quindi questi dati per imparare le relazioni. Questo passo richiede una certa quantità di tempo e risorse computazionali, ma è fondamentale per la creazione di modelli accurati. Una volta addestrato, il modello deve essere **valutato** per determinare quanto bene si comporta su dati in esso non presenti.

Questo passaggio è cruciale per capire se il modello è in grado di generalizzare bene oltre i dati con cui è stato addestrato.

Se il modello non soddisfa le prestazioni desiderate, può essere necessario **ottimizzarlo** regolando vari parametri o utilizzando tecniche di regolazione. Questo processo è noto come ottimizzazione del modello e può richiedere un po' di sperimentazione per trovare la combinazione ottimale di parametri.

MEDIAPIPE

Questo grosso lavoro di creazione dell'algoritmo, addestramento ed ottimizzazione è fornito in modo Open Source da Google con la potentissima piattaforma MediaPipe.

Il significato del termine "MediaPipe" è legato al concetto di "pipeline dei media". Si potrebbe brutalmente tradurre il termine "pipeline" come condotta o tubatura, nel contesto informatico lo accostiamo al concetto di **flusso di dati** e del consequenziale loro processo. Abbiamo infatti detto che i dati sono i mattoni portanti dei sistemi di ML. Il termine media è nel nostro parlare quotidiano, quindi con esso identifichiamo il mezzo che gestisce questo flusso di dati.

MediaPipe viene definita da Google on-device, cioè anche capace di funzionare su un dispositivo hardware senza necessità di collegamento internet. Ciò rende i nostri progetti portatili ed integrabili in svariati contesti.

Google aggiunge il termine "no-code" alla sua piattaforma, cioè fornisce un sistema che necessita di poca conoscenza della programmazione informatica per raggiungere il suo obiettivo. Concetto che conosciamo bene con l'universo Arduino, che ha aperto il mondo dell'elettronica e della scrittura del codice informatico anche a chi non ne aveva le basi. Ci sentiamo di dire che anche MediaPipe mira in questa direzione, proponendo in modo semplice un argomento in origine molto complesso come l'intelligenza artificiale.

Vediamo subito le potenzialità andando all'indiriz-

zo internet: https://mediapipe-studio.webapps.google.com/studio/demo/gesture_recognizer.

Occorre effettuare il login con un account Google. Diamo il consenso all'uso della web cam da parte del browser.

Inquadrando la nostra mano, destra o sinistra è indifferente, poiché identifica anche quale sia ed eseguiamo uno dei seguenti gesti:

- 1 – pugno chiuso
- 2 – palmo aperto
- 3 – indice che punta in su
- 4 – pollice in su
- 5 – pollice in giù
- 6 – simbolo della vittoria con due dita
- 7 – ti amo nella lingua dei gesti (posizione finale).

La pagina mostrerà, in tempo reale e sovrascritto al video, 21 punti chiave del palmo della mano collegati tra di loro mediante segmenti colorati **Fig.1**.

I punti chiave vengono definiti anche come **landmark** e rappresentano le parti significative delle mani, come le punte delle dita, i pollici, i palmi e così via. Il tracciamento preciso di questi landmark

Fig.1
Esempio di riconoscimento del gesto pollice su.

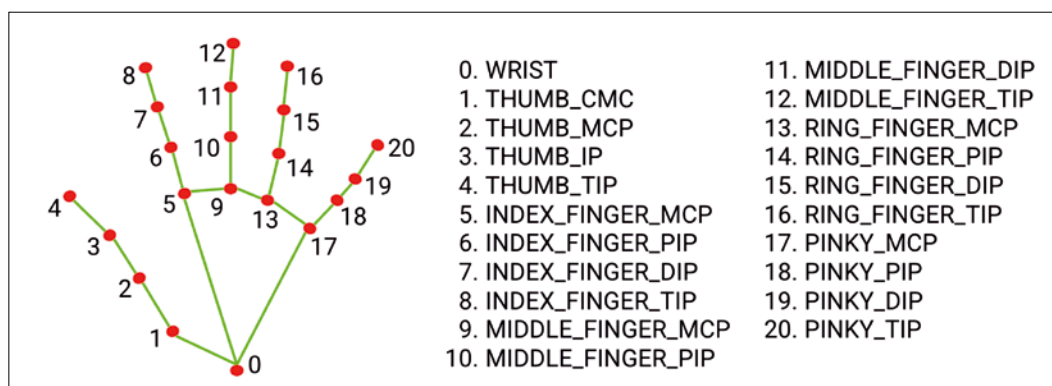
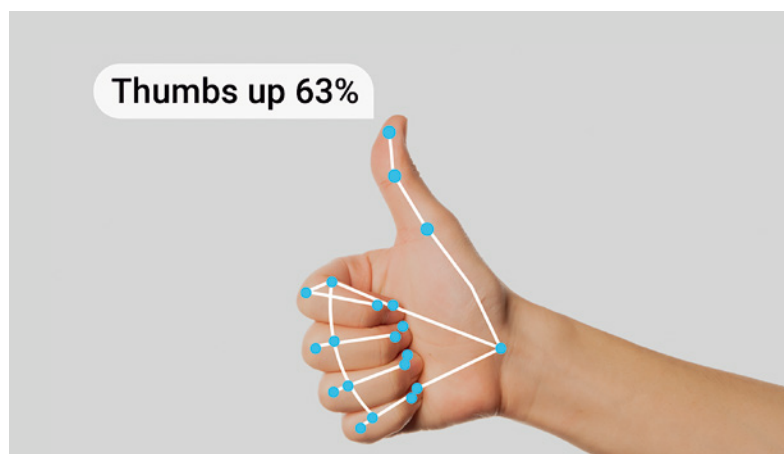


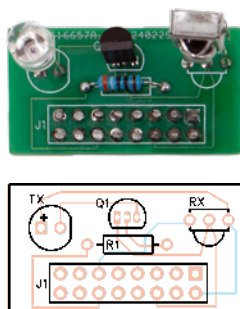
Fig. 2
Punti riconosciuti sul palmo della mano - landmark.

Elenco Componenti:

R1: 390 ohm
Q1: 2N3904
OPTO1: TSAL7600
U1: IR38DM

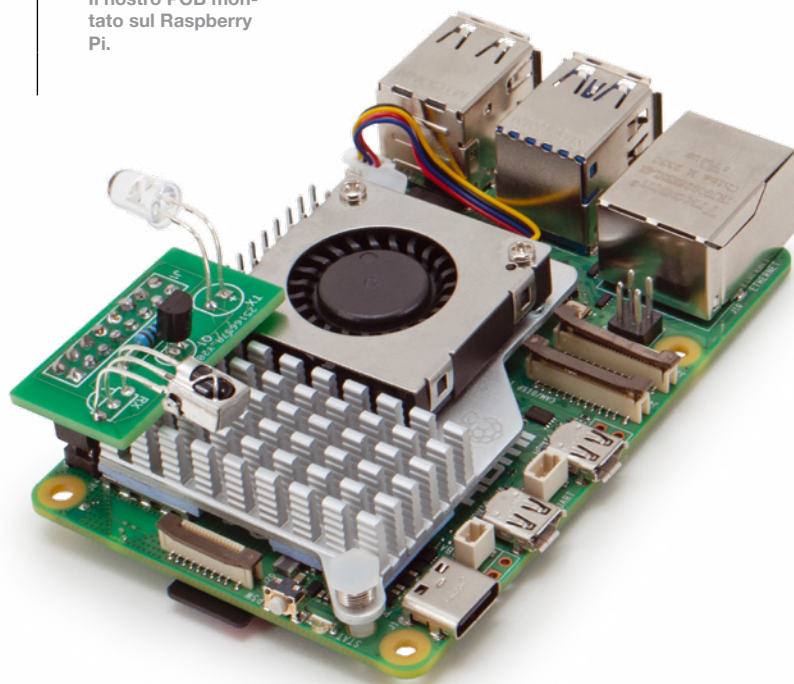
Varie:

- Strip Femmina 2x8 vie passo 2.54 mm
- Circuito Stampato S1749 (17x31 mm)



consente di comprendere la posa e i movimenti delle mani nell'ambito di un'applicazione (**Fig.2**). Nella pagina web, sotto il riquadro del video, vi saranno anche una serie di informazioni aggiuntive, come la percentuale di sicurezza nel aver riconosciuto il gesto eseguito ed ovviamente il tipo di gesto. MediaPipe offre quindi una soluzione chiamata "Hand Tracking" (tracciamento delle mani) che consente il riconoscimento e il tracciamento delle mani in tempo reale all'interno di un flusso video. Questo modulo di MediaPipe è stato denominato di conseguenza "HandLandmarks". Il modulo inizia rilevando la presenza di mani nel frame video e identifica le regioni dell'immagine che contengono le mani. Una volta rilevate le mani, inizia a tracciare i landmark delle mani.

Fig. 3
Il nostro PCB montato sul Raspberry Pi.



Con i landmark delle mani identificati, il modulo può stimare la posa delle mani all'interno del frame video. Questo include informazioni come l'orientamento delle mani, la direzione delle dita e altri dettagli che possono essere utilizzati per interpretarne i gesti.

MediaPipe è basata su TensorFlow lite, cioè la grande piattaforma di ML sempre di Google, ma ne differisce nel fatto che è già pre-addestrato.

Il riconoscimento gestuale sfrutta ad esempio circa 30000 immagini prese dal mondo reale.

Potrete anche addestrare direttamente voi il sistema, con immagini di altri gesti seguendo il tutorial al seguente link: https://developers.google.com/mediapipe/solutions/customization/gesture_recognizer.

Divertitevi ad esplorare gli altri sistemi di ML forniti da MediaPipe come ad esempio la

classificazione delle immagini, rilevamento degli oggetti o dei volti. La documentazione ivi presente

vi permetterà di approfondire gli argomenti più di interesse. MediaPipe è anche multi piattaforma,

infatti si possono creare progetti per Android, iOS ed Web (JavaScript). Nel nostro progetto useremo

Python cioè uno dei linguaggi di programmazione più popolari nel campo del ML (e non solo) per

diverse ragioni: la sua semplicità d'uso, la sintassi intuitiva e la vasta gamma di risorse didattiche e

librerie disponibili lo rendono un'ottima scelta per i principianti e gli esperti di programmazione.

Un articolo non basterebbe neanche come pre-

messaggio per imparare il grande mondo di TensorFlow, MediaPipe ed Python. Ma vorremo che vi

servisse per capirne le capacità e vi permettesse di indirizzare lo studio su argomenti indispensabili nel

campo dell'informatica presente e futura.

MEDIAPIPE SU RASPBERRY PI

Entriamo allora nel pratico ed installiamo MediaPipe sul nostro Raspberry Pi 5, in modo da utilizzarlo subito on-device. Collegatevi alla vostra board come meglio siete abituati (noi consigliamo VNC, per semplicità e potenza di utilizzo, troverete un tutorial sul nostro sito <https://futuranet.it/progetti/2023/04/20/realizzare-un-sistema-di-sorveglianza-con-raspberry-pi-e-opencv/>).

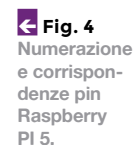
Python è nativamente presente nel sistema operativo di Raspberry Pi. Come sempre prendete i diritti da amministratore con il comando `sudo bash` e accertatevi di avere tutto aggiornato con `apt update` ed `apt upgrade`. Installiamo la libreria MediaPipe per Python con il comando da terminale:

```
pip install mediapipe
```


Installiamo con: *pip install opencv-python*
Abbiamo tutto per provare il nostro sistema
stand-alone (applicazione o sistema che può
funzionare in modo indipendente). Se siete già

Da notare come nella sezione *mache-case* abbiamo lasciato come commento (in Python simbolo # all'inizio della riga) le istruzioni *os.system*, cioè quei comandi da sistema operativo che permetteranno di inviare i segnali IR, che renderemo operativi dopo aver trattato l'aspetto hardware.

- $2 \rightarrow V_{CC} + 5V$



- 6 → Gnd
- 11 → GPIO 17 per il ricevitore IR.
- 12 → GPIO 18 per il trasmettitore IR.

Abbiamo usato un semplice transistor per alimentare a 5Vcc il led IR in modo da dargli più potenza, garantendo la tutela dei GPIO di Raspberry Pi funzionanti a 3.3Vcc.

Il trasmettitore usato è un TSAL7600, un diodo che emette infrarossi a 940 nm ad alto potere radiante, incorporato in un involucro di plastica trasparente. Se avete a disposizione qualche vecchio telecomando potrete provare anche con il suo led IR.

Il ricevitore è un IR38DM (anche trovabile come MIM-5385K2), sensibilissimo sensore IR con preamplificatore, demodulatore e filtro tarato sulla frequenza di 38 kHz – 600 µs con amplificatore/squadratore incorporato (**Fig.5**).

Non ci addentreremo troppo nella trattazione dei segnali IR, la nostra rivista ne ha più volte approfondito lo studio, si consiglia a riguardo l'articolo "Decoder per telecomandi IR" del numero 247 del Settembre 2020.

Potrete acquistare riviste arretrate e componenti sul nostro store www.futurashop.it.

L'approccio utilizzato è quello di memorizzare su

Raspberry Pi il segnale decodificato dal ricevitore, per poi rimandarlo in uscita con il trasmettitore, abbinando ad ogni gesto un segnale di uscita. La decodifica effettuata dal sensore ed una libreria aggiuntiva installata sulla board permetteranno di avere una sequenza di numeri, che memorizzeremo in semplice file di testo (quelli richiamati dai comandi `os.system`), per ogni tasto del telecomando che vorremmo implementare.

Raspberry Pi ha una funzionalità nativa per abilitare la trasmissione e la ricezione dei file dei segnali IR. Apriamo per la modifica il file `/boot/firmware/config.txt`.

Per chi non fosse pratico con RaspberryOS, consigliamo di effettuare tutte le istruzioni di modifica dei file con l'editor di sistema nano. Esso, a differenza degli editor da GUI, se aperto da terminale con in diritti di amministratore (`sudo bash`) permette l'apertura, la modifica e il salvataggio di tutti i tipi di file, anche di sistema. Quindi da terminale scriviamo:

```
nano /boot/firmware/config.txt
(fate attenzione alla corretta patch)
```

Togliere i commenti (anche qui come in Python il carattere #) per abilitare le seguenti righe sotto

```
# Additional overlays and parameters are documented:
dtoverlay=gpio-ir,gpio_pin=17
dtoverlay=gpio-ir-tx,gpio_pin=18
```

Su alcune versioni di RaspberryOS potreste proprio non trovarle, dovete in questo caso semplicemente scriverle voi. Quindi stiamo dicendo alla configurazione del sistema operativo di impostare il pin 12 (GPIO18) come trasmettitore IR ed il pin 11 (GPIO17) come ricevitore IR.

Questo è possibile perché tali funzionalità sono native nel kernel Linux.

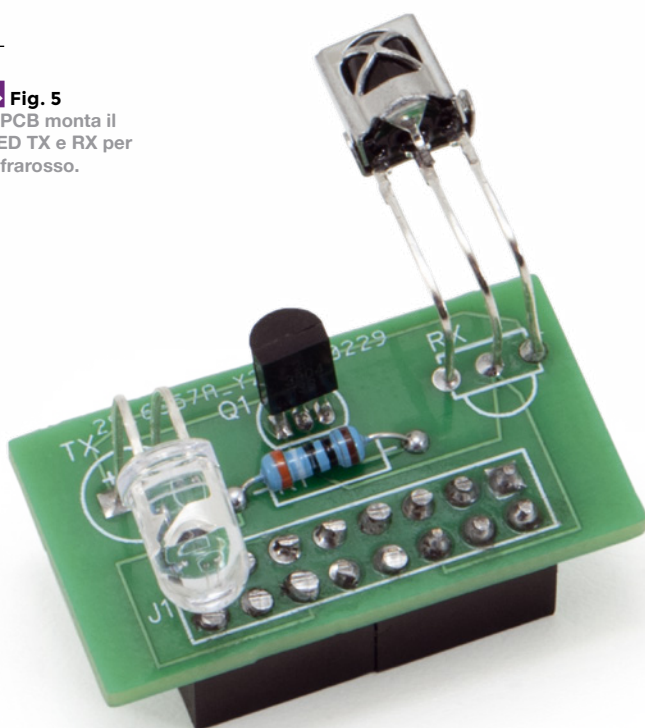
Riavviamo per rendere effettive le configurazioni. Ora occorre installare una funzionalità fondamentale per implementare in nostro sistema.

Da terminale:

```
apt install v4l-utils
```

Le "v4l-utils" sono un pacchetto di utility open-source progettate per lavorare con dispositivi video Linux (V4L, Video for Linux). V4L è un sottosistema del kernel Linux che consente alle applicazioni di interagire con dispositivi video, come webcam, sintonizzatori TV, schede di acquisizione video, ed

Fig. 5
Il PCB monta il LED TX e RX per infrarosso.



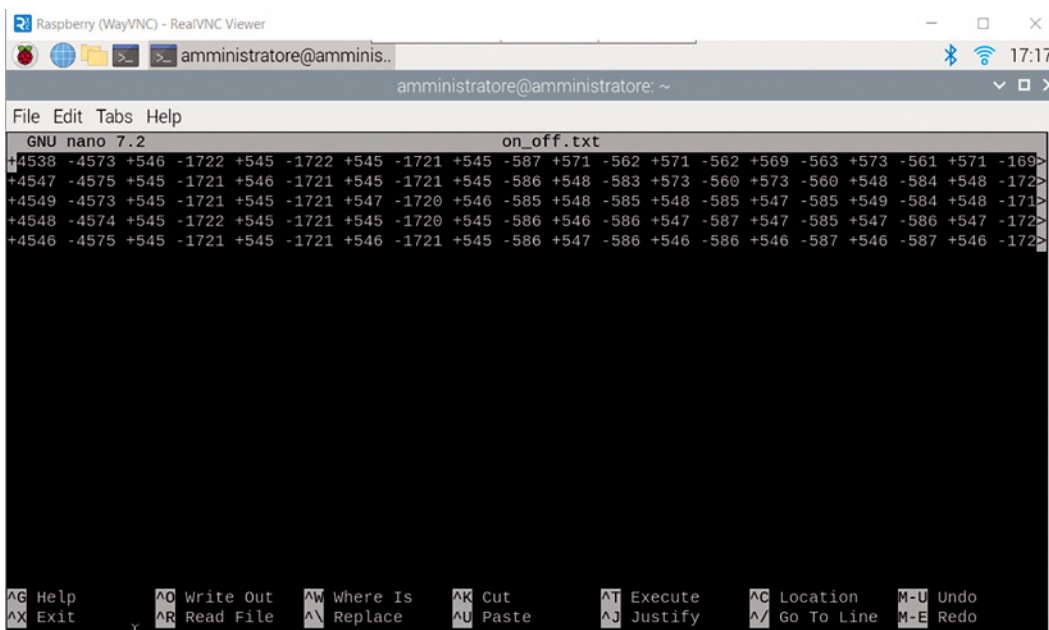


Fig. 6
Segnale ricevuto dal telecomando relativo al tasto on-off in codifica mode2.

ovviamente segnali IR.

Le utility di *v4l-utils* offrono un set di strumenti per configurare, controllare e gestire questi dispositivi.

Nel **Listato 1** avete notato come abbinato al comando *os.system* (cioè istruzione Python che esegue direttamente una funzionalità appartenente al sistema) vi sia l'istruzione "**ir-ctl**".

Essa è una delle utility presenti all'interno di *v4l-utils* che consente di controllare dispositivi IR. Con *ir-ctl*, è possibile inviare e ricevere segnali infrarossi.

Nella cartella */dev/* sono presenti il file *lirc0* per l'invio dei segnali e *lirc1* per la trasmissione, che sono funzionalità derivate dal pacchetto *LIRC* ormai integrato nel kernel del SO nelle ultime versioni di Linux (per approfondire *lirc.org*).

Mettiamolo subito alla prova scrivendo:

```
ir-ctl -d /dev/lirc1 --receive=on_off.txt
```

Il terminale rimarrà in attesa di eventi. Armatevi

del telecomando che volete "clonare", premete il tasto on-off direzionando lo stesso verso il ricevitore IR. Cercate di premerlo velocemente ed una volta sola, in modo da non inviare troppi codici contemporaneamente. Premete *ctrl+c* per terminare l'acquisizione. Analizziamo il file creato con *nano on_off.txt*.

Il file dovrebbe apparire come in **Fig. 6**.

La sequenza numerica ricevuta è in "*mode2*" cioè una specifica tipologia di codifica dei segnali IR catturati.

Ogni coppia di numeri rappresenta le lunghezze degli impulsi (+ o pulse) e gli intervalli tra gli impulsi (- o space). Anche il *mode2* è una funzionalità di *lirc*, come già detta ormai integrata nel SO (<https://manpages.ubuntu.com/manpages/jammy/man1/ir-ctl.1.html> per approfondire).

Se avete premuto per troppo tempo il tasto on_off potreste aver memorizzato più righe di codici *mode2*. Dovrete quindi cancellare quelli che si ripetono, infatti della codifica basta la prima riga,

GESTO DELLA MANO	COMANDO TV	FILE TXT
PALMO APERTO	ON OFF	on_off.txt
POLLICE SU	VOLUME SU	volume_su.txt
POLLICE GIÙ	VOLUME GIÙ	volume_giu.txt
DITO IN SU	CANALI AVANTI	canali_avanti.txt
VITTORIA	CANALI INDIETRO	canali_indietro.txt

Tabella 1
Corrispondenze del gesto della mano con i comandi TV.

▼ Listato 1

```
# Import delle librerie necessarie
import os
import cv2
import mediapipe as mp
from mediapipe.tasks import python

# Numero massimo di mani da rilevare e percorso del modello di riconoscimento dei gesti
numero_mani = 1
model_path = "gesture_recognizer.task"

# Inizializzazione della libreria MediaPipe
GestureRecognizer = mp.tasks.vision.GestureRecognizer
GestureRecognizerOptions = mp.tasks.vision.GestureRecognizerOptions
GestureRecognizerResult = mp.tasks.vision.GestureRecognizerResult
VisionRunningMode = mp.tasks.vision.RunningMode

# Variabile per memorizzare l'ultimo gesto riconosciuto
last_gesture = None

# Funzione per stampare il risultato del riconoscimento dei gesti
def print_result(result: GestureRecognizerResult, output_image: mp.Image, timestamp_ms: int):
    global last_gesture

    for single_hand_gesture_data in result.gestures:
        gesture_name = single_hand_gesture_data[0].category_name
        # Controlla se il gesto riconosciuto è diverso dall'ultimo gesto rilevato
        # Serve per dare un impulso ir dopo ogni gesto
        if gesture_name != last_gesture:
            last_gesture = gesture_name
            # Stampa il nuovo gesto riconosciuto
            print("Nuovo gesto riconosciuto:", gesture_name)
            # Corrispondenze tra il gesto riconosciuto e le azioni da eseguire
            # Quando avrete a disposizione i comandi ir decommentate le istruzioni os.system
            match gesture_name:
                case "Open_Palm":
                    print("palmo aperto")
                    # os.system('ir-ctl -d /dev/lirc0 --send=on_off.txt')
                case "Thumb_Up":
                    print("pollice su")
                    # os.system('ir-ctl -d /dev/lirc0 --send=volume_su.txt')
                case "Thumb_Down":
                    print("pollice giu")
                    # os.system('ir-ctl -d /dev/lirc0 --send=volume_giu.txt')
                case "Pointing_Up":
                    print("dito in su")
                    # os.system('ir-ctl -d /dev/lirc0 --send=canali_avanti.txt')
                case "Victory":
                    print("vittoria")
                    # os.system('ir-ctl -d /dev/lirc0 --send=canali_indietro.txt')

# Opzioni per il riconoscitore dei gesti
options = GestureRecognizerOptions(
    base_options=python.BaseOptions(model_asset_path=model_path),
    running_mode=VisionRunningMode.LIVE_STREAM,
    num_hands = numero_mani,
    result_callback=print_result)

# Creazione del riconoscitore dei gesti
recognizer = GestureRecognizer.create_from_options(options)

timestamp = 0
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands

# Inizializzazione del riconoscitore delle mani
hands = mp_hands.Hands(
    static_image_mode=False,
    max_num_hands=numero_mani,
    min_detection_confidence=0.65,
    min_tracking_confidence=0.65)

# Inizializzazione della cattura video dalla webcam
```



```

# Se avete più dispositivi di input video collegati a Raspberry
# occorre mettere il numero giusto al posto dello 0
cap = cv2.VideoCapture(0)

# Ciclo principale del programma
while True:
    # Acquisizione di un frame dalla webcam
    ret, frame = cap.read()
    if not ret:
        break

    # Flip orizzontale del frame
    frame = cv2.flip(frame, 1)
    # Conversione del frame da BGR a RGB (compatibile con MediaPipe)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    # Elaborazione del frame per il rilevamento delle mani
    results = hands.process(frame)
    # Conversione del frame da RGB a BGR per la visualizzazione
    frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
    # Creazione di un array numpy con il frame convertito in RGB
    np_array = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Se vengono rilevate delle mani nel frame
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            # Disegno dei landmark della mano sul frame
            mp_drawing.draw_landmarks(frame, hand_landmarks, mp_hands.HAND_CONNECTIONS)
            # Creazione di un oggetto di tipo Image compatibile con MediaPipe
            mp_image = mp.Image(image_format=mp.ImageFormat.SRGB, data=np_array)
            # Riconoscimento asincrono dei gesti sulla mano
            recognizer.recognize_async(mp_image, timestamp)
            timestamp = timestamp + 1 # Deve aumentare monotonamente in modalità LIVE_STREAM

    # Mostra il frame con i gesti rilevati
    cv2.imshow('Riconoscimento Gestuale', frame)

    # Se il tasto Esc viene premuto, interrompi il ciclo
    if cv2.waitKey(1) & 0xFF == 27:
        break

# Rilascia la cattura video e chiudi tutte le finestre
cap.release()
cv2.destroyAllWindows()

```

le altre sono praticamente ripetizioni della codifica ricevuta (si nota infatti la somiglianza tra le righe). Dopo aver cancellato le righe non necessarie,

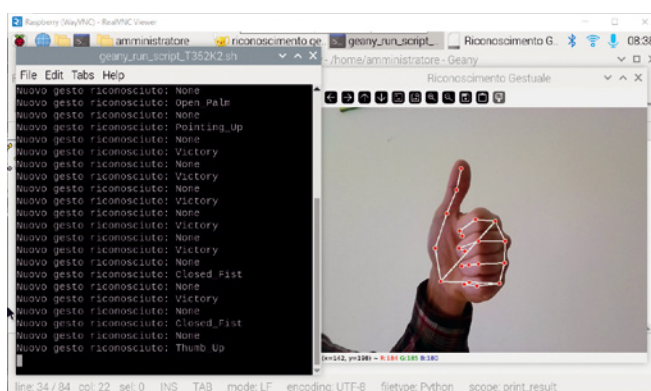
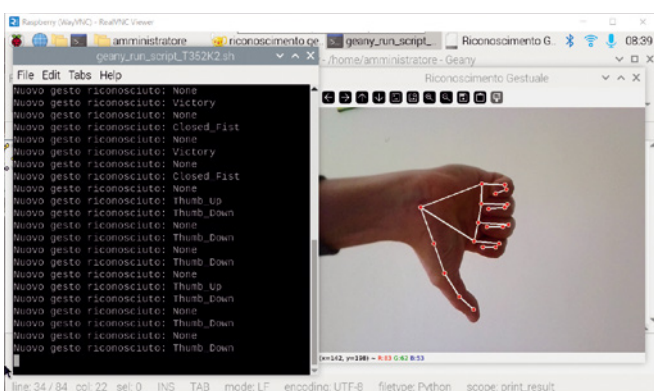
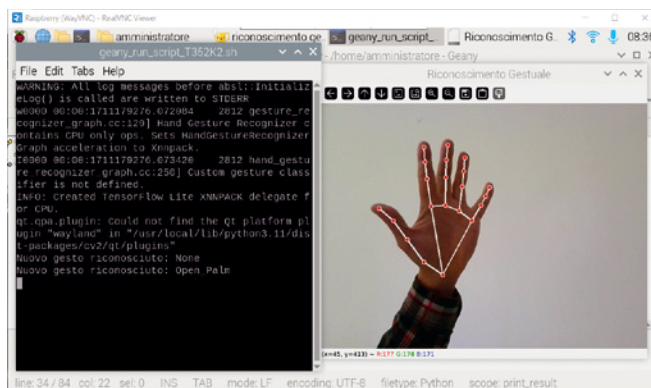
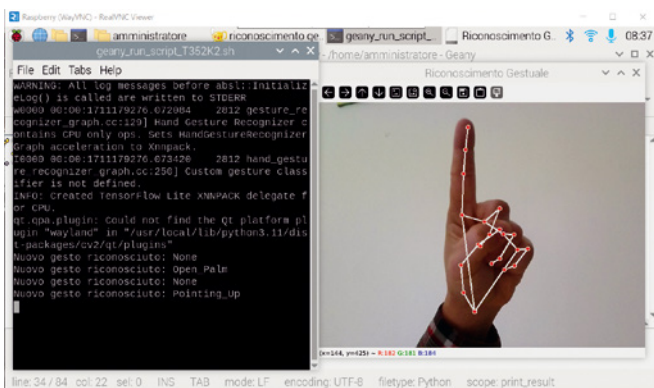
andiamo alla fine del primo e cancelliamo l'ultimo valore, che rappresenta il timeout tra qui invii di segnali (tra le righe). Tale valore non serve nella

```

Raspberry (WayVNC) - RealVNC Viewer
amministratore@amminis...
amministratore@amministratore: ~
File Edit Tabs Help
amministratore@amministratore:~$ sudo bash
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=volume_su.txt
^C
root@amministratore:/home/amministratore# nano volume_su.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc0 --send=volume_su.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=volume_giu.txt
^C
root@amministratore:/home/amministratore# nano volume_giu.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=canali_avanti.txt
^C
root@amministratore:/home/amministratore# nano canali_avanti.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc0 --send=canali_avanti.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc1 --receive=canali_indietro.txt
^C
root@amministratore:/home/amministratore# nano canali_indietro.txt
root@amministratore:/home/amministratore# ir-ctl -d /dev/lirc0 --send=canali_indietro.txt
root@amministratore:/home/amministratore#

```

Fig. 7
Registrazione ed invio segnali IR telecomando.



Il nostro sistema in funzione.

trasmissione, ed è facilmente identificabile anche perché è un numero molto grande rispetto agli altri. Salvate il file e tornate al terminale (`ctrl+x` ; `yes`). Indirizzate il trasmettitore verso il televisore e scrivete da terminale:

```
ir-ctl -d /dev/lirc0 --send=on_off.txt
```

Se tutto ha funzionato il vostro TV si sarà "magicamente" acceso (o spento se fosse già acceso).

Eseguite la stessa procedura per gli altri tasti

Fig.7, potete ovviamente modificare a piacere le corrispondenze. Quelle del nostro progetto sono riportate in **Tabella 1**.

Consigliamo di posizionare il file Python, il modello MediaPipe (gesture_recognizer.task) ed i file txt nella cartella principale home (nel nostro progetto / home/amministratore).

Mandate in esecuzione il file Python, e posizionatevi nell'inquadratura della webcam. Alternate i comandi gestuali con il pugno chiuso, che non ha istruzioni abbinare, in modo da discriminarli in modo adeguato. Aprendo il palmo vi siete sentiti un po' Jedi?

CONCLUSIONE

Abbiamo iniziato a esplorare il mondo dell'intelligenza artificiale e del machine learning attraverso un progetto semplice, che tuttavia può consentire la costruzione di un percorso di studi su un ambito nuovo e affascinante, verso il quale la tecnologia moderna si sta rapidamente dirigendo. Anche nel campo dei maker, Raspberry Pi offre la possibilità di sostituire la nostra PCB con qualsiasi dispositivo di output, consentendo di controllare, ad esempio, i carichi elettrici della nostra abitazione.



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
RPI5-4GB	Raspberry Pi 5 versione con 4 GB di memoria	105,00
RPI5-8GB	Raspberry Pi 5 versione con 8 GB di memoria	136,00
LED5IR	LED TX infrarosso 5mm	0,60
IR38DM	Sensore ricevitore IR	2,50

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

La porta d'accesso al mondo dell'elettronica!



SFRUTTA AL MEGLIO L'ENERGIA DEL SOLE!



**600
WATT**

cod. FR861

€ 340,00

MICROINVERTER SOLARE WIFI AD IMMISSIONE DIRETTA IN RETE

Inverter di tipo "Plug&Play" ideale per realizzare mini impianti fotovoltaici, dove si desidera immettere direttamente in rete l'energia prodotta dall'impianto solare, andando a ridurre l'energia prelevata. Con una potenza massima di picco di 450W per canale (due canali disponibili), è in grado di gestire elevati livelli di potenza. Inoltre, grazie al WiFi integrato, è possibile monitorare il microinverter da remoto tramite un'app gratuita (NEPViewer), consentendo una gestione del sistema fotovoltaico semplice e intuitiva.

Risponde alla norma CEI 0-21 per la realizzazione di mini impianti fotovoltaici.

ATS (AUTOMATIC TRANSFER SWITCH) COMMUTATORE AUTOMATICO DI ALIMENTAZIONE

L'ATS è progettato per commutare automaticamente tra l'alimentazione fornita dai pannelli solari e quella proveniente dalla rete elettrica tradizionale. Se l'energia solare è sufficiente per coprire le esigenze energetiche della casa, l'ATS utilizza automaticamente l'elettricità prodotta dai pannelli solari. Nel caso in cui la produzione di energia elettrica solare non sia sufficiente, l'ATS commuta automaticamente sulla rete elettrica tradizionale garantendo in questo modo l'alimentazione senza interruzioni.



cod. FR863

€ 194,00

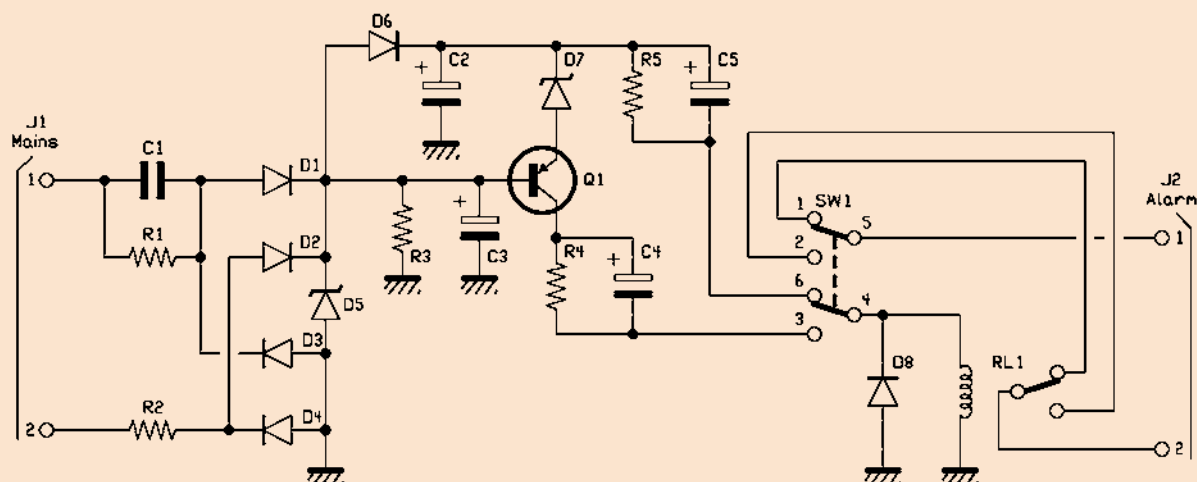
MONITOR ALLARME DI ASSENZA RETE ELETTRICA



di **STEFANO PURCHIARONI**

Scoraggiamo i malintenzionati che ci staccano la corrente, con un dispositivo che attiva l'allarme se manca la tensione di rete per più di qualche secondo, con conteggio degli eventi occorsi.

Presentiamo due soluzioni circuitali, una analogica ed una digitale con micro-controllore, per sorvegliare la presenza della rete elettrica presso l'utenza domestica. Questa funzione parte da un'esigenza di protezione contro eventuali malintenzionati che potrebbero disconnettere la rete elettrica agendo sul contatore esterno all'abitazione, con lo scopo di penetrarvi una volta lasciate scaricare le batterie degli impianti di allarme. Attivare gli avvisatori acustici già al momento della disconnessione aiuterà a rendere la casa più sicura. Per evitare attivazioni improprie, è stato previsto un tempo di tolleranza di qualche secondo, che eviterà allarmi inutili in caso di brevi interruzioni. In aggiunta a questa funzione di base, la versione digitale



conta anche gli eventi di caduta della tensione di rete, consente una regolazione dei tempi di allarme e tolleranza, ed inoltre spegne la sirena una volta esaurito il tempo di allarme prefissato. Oltretutto, agendo su due dip-switch presenti sullo stampato, è possibile attivare una funzione "Beep" e una segnalazione di "Ritorno della rete elettrica". Torniamo più avanti a queste funzioni aggiuntive.

DIFFERENZE TRA LA VERSIONE ANALOGICA E QUELLA DIGITALE

Versione Analogica

Come si vede dallo schema, la sezione di alimentazione è priva di trasformatore, che troveremo invece in quella digitale. Ciò comporta una maggiore attenzione da porre durante i test, perché tutta la circuiteria è sottoposta alla tensione di rete, con pericolo di folgorazione in caso di contatto. Nel caso della versione digitale tale pericolo sussiste solo sulla piccola parte di circuito connessa al primario del trasformatore, cosa che comunque

impone un alto livello di attenzione durante le fasi di test.

Dopo la sezione di raddrizzamento dell'alternata e limitazione della tensione a 24 volt, costituita dai diodi D1..D4 e dallo Zener D5, notiamo due derivazioni di tale sorgente di tensione. La serie D6-C2 produce 23.3 volt stabilizzati e livellati, utilizzati per attivare il relè tramite il percorso 4-6 dello switch a due posizioni. Quando la rete viene a mancare, il relè torna nella posizione di riposo chiudendo così il contatto COM-NC. Ciò attiva l'allarme con circa 3-4 secondi di ritardo a causa del protrarsi dell'erogazione di energia tramite la carica immagazzinata dal grosso elettrolitico C2. Al ritorno dell'energia elettrica, la disattivazione dell'allarme avverrà con circa 2 secondi di ritardo, tempo impiegato da C2 per caricarsi nuovamente e fornire una tensione abbastanza alta da poter alimentare di nuovo il relè. Il condensatore C5 inizialmente scarico permette di attivare istantaneamente il relè a piena tensione ma, una volta carico, la corrente per il suo sostentamento viene fornita attraverso R5, e ridotta

Tabella. 1
Funzioni disponibili
nelle due versioni.

FUNZIONE	ANALOGICA	DIGITALE
Tolleranza nei confronti di brevi interruzioni	X	X
Regolazione del tempo di tolleranza	-	X
Interruzione dell'allarme dopo un tempo prefissato	-	X
Regolazione del tempo di allarme	-	X
Conteggio e visualizzazione del numero di cadute di tensione	-	X
Attivazione impulsiva impostabile in alternativa a quella permanente	X	X
Segnalazione dell'avvenuto ritorno dell'energia elettrica	-	X

leggermente restando tuttavia ad un livello più che sufficiente per sostenere l'ancoraggio del relè. Questa tecnica allunga notevolmente la vita utile di tale componente, che resta leggermente sottoalimentato per la maggior parte del tempo. Il relè deve essere a due scambi, con bobina a 24 volt e resistenza interna di 1600 ohm. Un eventuale distacco della rete elettrica che duri meno del tempo di ritardo prima menzionato non produrrà l'attivazione dell'allarme, poiché C2 non si sarà scaricato abbastanza da far scendere la tensione fornita al relè al di sotto della sua Voff, misurata sperimentalmente attorno ai 4 volt. Questa tolleranza nei confronti delle brevi interruzioni permette di evitare inutili allarmi. In questa versione Analogica, il tempo di tolleranza non è regolabile. Un aspetto negativo di questa versione riguarda il tempo di allarme. Una volta scaricato C2 il relè si spegnerà definitivamente, lasciando l'allarme attivo indefinitamente. La gestione di un tempo massimo dovrà essere implementata a valle del circuito, o facente parte della sirena o altro dispositivo scelto. Nel caso in cui lo switch SW1 sia posizionato in modo da chiudere i contatti 5-2 e 4-3, il relè verrà alimentato da C2, in assenza di rete, solo dopo che C3 si scaricherà fino a portare la base di Q1 ad un livello di tensione sufficientemente basso da portarlo in conduzione, lasciando così defluire la corrente da C2 verso il relè passando per C4. Dopo circa mezzo secondo quest'ultimo si sarà caricato, riducendo la circolazione di corrente tanto da abbassare la tensione sulla bobina del relè sotto la sua soglia di tenuta, spegnendolo. I valori dei condensatori e delle resistenze sono già calcolati per far sì che la carica di C2 possa sostenere il ciclo. Al ritorno dell'energia elettrica, Q1 tornerà in interdizione, il relè non muterà il suo stato, e C2 sarà nuovamente carico ed in grado di supportare nuovi cicli. Da osservare l'importante funzione svolta dal diodo D6 in questa modalità, che agisce come "valvola di non ritorno" evitando che la base di Q1 resti alta a causa della carica accumulata in C2, col risultato di veder impedita l'attivazione del relè. I due modi di funzionamento sono opposti. Infatti nella modalità standard il relè è sempre attivo, mentre nella modalità ad impulso, esso si attiva solo brevemente. Per ottenere comunque l'attivazione dell'allarme si utilizza il secondo scambio dello switch, tramite i suoi contatti 5-1 per l'attivazione permanente, oppure 5-2 per quella ad impulso. Il circuito stampato prevede di remotizzare lo switch e, nel caso occorra invertire la logica

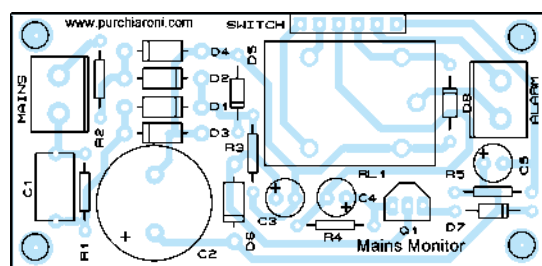
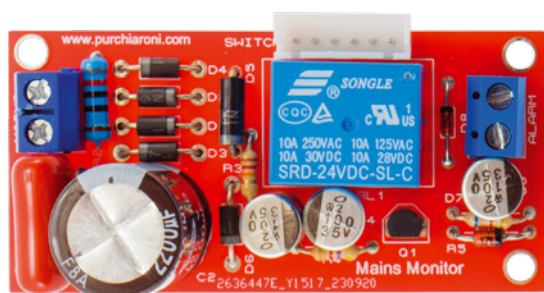
di attivazione dell'allarme ad "Apertura del circuito" anziché "Chiusura", occorre invertire i fili destinati ai contatti 1 e 2 dello switch stesso.

Versione Digitale

Notiamo subito la maggior complessità dello schema. L'alimentazione è fornita tramite un trasformatore con secondario da 6 volt, che alimenta due ponti raddrizzatori. BR2, seguito dal gruppo di stabilizzazione costituito dal condensatore C3, dal piccolo regolatore lineare U1 da 5 volt e dal condensatore C4 posto alla sua uscita, fornisce l'alimentazione per il microcontrollore U2, un versatile PIC16F1827. In caso di assenza di tensione di rete, il supercondensatore C5 da 1,5F continuerà a fornire una tensione sufficiente ad alimentare il PIC per un tempo considerevole. La gestione del supercondensatore è ottenuta tramite i diodi di blocco D2, D3 e D4 che lo isolano durante il funzionamento nominale, garantendone però la ricarica. Da notare il jumper JP1, previsto per l'eventuale rimozione e riprogrammazione del PIC senza che resti pericolosamente alimentato dal supercondensatore. Per evitare eccessive cadute di tensione dovute ai diodi D2 e D4, per questi è stato selezionato un modello "Shottky" che ha una tensione di soglia diretta (Vds) di soli 0,2 volt. Questo permette di caricare C5 fino a 4,8 volt, ed ottenere 4,6 volt per l'alimentazione di backup del PIC, che ha una tensione minima di funzionamento di soli 2,8 volt per il modello scelto.

La riduzione del consumo del microcontrollore, durante la fase di alimentazione da supercondensatore, è ottenuta attivando via software la modalità "Nanowatt" introdotta dalla Microchip sulla maggior parte dei suoi microcontrollori. Essa pone il componente in uno stato di basso consumo dove solamente l'oscillatore da 32.768 Hz resta in funzione, riattivando l'esecuzione del codice solamente in corrispondenza agli interrupt provenienti dal Timer associato a tale oscillatore. Gli interrupt risvegliano quindi l'esecuzione del programma di gestione ogni secondo, sostanzialmente per contare i secondi trascorsi e decidere se è arrivato il momento di agire sul relè, confrontando il contasecondi con la lettura dei due trimmer resistivi RV1 e RV2. Una ulteriore riduzione nel consumo di energia è resa possibile collegando il lato caldo dei due trimmer resistivi non direttamente alla tensione di alimentazione, ma al pin digitale di output RA2, che verrà posto al livello logico alto solo durante la lettura dei due trimmer, effettuata tramite gli ingressi analogici AN3 ed AN4.





Elenco Componenti:

R1: 1 Mohm 1/2 W
R2: 820 ohm 1/2 W
R3: 18 kohm
R4: 4,7 kohm
R5: 680 ohm
C1: 470 nF 450 VL poliestere
C2: 2200 µF 35 VL elettrolitico

C3, C4, C5: 100 µF 35 VL
elettrolitico
Q1: BC557
D1, D2, D3, D4, D6: 1N4007
D8: 1N4148
D5: Zener 24 V / 1 W
D7: Zener 5,1 V / 0,5 W

RL1: Relè 24 V DPDT 3 A
SW1: Deviatore DPDT
J1, J2: Morsetto 2 poli passo 5 mm

Varie:
- Circuito Stampato S1722 (71x34 mm)

Il rilevamento della caduta della tensione di rete, è affidato alla sezione che parte da BR1, priva di condensatori di alta capacità a valle di esso, ma dotata di una circuiteria che genera 5,1V in presenza di energia. Al mancare di questa, la tensione di uscita targata SENSE scende molto rapidamente, comunicando l'evento al pin digitale di ingresso RB1 che interpreta come uno "Zero" binario una tensione inferiore a 0,8V, cosa che con i valori dei componenti di questa sezione avviene dopo circa mezzo secondo dalla caduta della tensione di rete. Montando la resistenza opzionale R3, è possibile ridurre ulteriormente questo ritardo. Sulla base del livello Alto o Basso del pin RB1, il software di gestione attiva l'allarme, secondo i tempi impostabili dai due trimmer resistivi RV1 e RV2 che permettono di regolare il tempo di tolleranza tra 0 e 10 secondi, ed il tempo di allarme da 0 a 2 minuti.

Come accennato prima, il conteggio dei secondi è affidato all'oscillatore interno a basso consumo del PIC, munito del piccolo quarzo esterno per orologi X1 da 32.768 Hz. Sempre per minimizzare i consumi, il modello scelto per il relè consiste in un Bistabile da 3V a doppia bobina, con due scambi completi. Questi ultimi sono messi in parallelo per aumentare la portata del relè, che comunque non potrà superare i 5 Ampere. Le due bobine di attivazione e disattivazione del relè sono pilotate dai due transistor Q1 e Q2, le cui basi fanno capo

ai pin RA6 e RA7 del PIC, che potrà così attivare e disattivare il relè con brevi impulsi positivi sui suddetti pin di output. Il microcontrollore è anche connesso ad un Dip Switch doppio, che consente di attivare le due funzioni opzionali "Beep" e "Back". La prima limita l'accensione dell'allarme a circa un secondo, per fornire una breve ma intensa segnalazione a chi per esempio è in casa, avvisando della caduta di tensione senza disturbare il vicinato. La funzione "Back" prevede due brevi impulsi per segnalare il ritorno della tensione di rete.

Questa funzione è però limitata dal tempo massimo di erogazione dell'energia offerto dal supercondensatore C5, che è di circa un paio d'ore. A completare il circuito notiamo il pulsante SW1 ed il led D5. Quest'ultimo ha il compito di segnalare con brevi impulsi gli eventi di caduta che si sono verificati. Se per esempio l'interruzione di energia è avvenuta due volte, D5 eseguirà ciclicamente due lampeggi, seguiti da una pausa di circa 3 secondi. Per resettare il conteggio basta premere il pulsante "Reset" SW1. Questo pulsante esegue anche un riavvio completo del PIC, ed una disabilitazione forzata del relè, utile in caso di problemi. I due led D7 e D9 segnalano l'emissione del breve comando di attivazione e disattivazione del relè che, se non seguiti dai "Click" dello stesso, aiutano ad identificare un guasto di questo componente elettromeccanico.

SOFTWARE DI GESTIONE

Focalizziamo l'attenzione sulle procedure principali "SetAlm", "interrupt" e "main". La procedura "SetAlm" riportata di seguito provvede all'attivazione del dispositivo di allarme connesso ai terminali di uscita J2, tramite impulsi diretti al relè bistabile. Il tempo di persistenza dell'impulso alle bobine di comando del relè è definito dalla costante "RlyTim" attualmente posta a 200 millisecondi. Le bobine fanno capo ai pin RA6 e RA7 del microcontrollore, mediati dai transistor Q1 e Q2. I led D7 e D9 provvedono a mostrare l'attivazione e la disattivazione del relè:

```
void SetAlm() {  
  // Activate the alarm and count the event in EEPROM  
  
  // Activate the alarm  
  ALMOFF=0;  
  ALMON=1;  
  delay_ms(RlyTim);  
  ALMON=0;  
  curs = 0;          // Reset the seconds counter  
}
```

La seguente procedura "interrupt" costituisce l'entry-point della ISR (Interrupt Service Routine) del PIC. Essa viene invocata dall'interrupt generato dal Timer-1 allo scadere del secondo, anche in modalità a basso consumo.

Sul circuito stampato è prevista una uscita sul Test Point TP1 per verificare con un frequenzimetro o un oscilloscopio il funzionamento dell'oscillatore. TP1 viene invertito di livello ad ogni secondo, producendo un segnale da 0,5 Hertz. La massa per la misura è disposta sul Test Point GND.

Lo scopo principale della procedura di interrupt è quello di conteggiare i secondi trascorsi, sulla variabile a 32 bit "curs".

L'interrupt per il prossimo ciclo viene nuovamente abilitato a valle della procedura:

```
void interrupt() {  
  // Interrupt Service Routine.  
  // It is called every second by Timer1 overflow.  
  
  // ----- TMR1 -----  
  // Manage Timer1 overflow (each 1s) to count seconds  
  if (PIR1.TMR1IF == 1) {  
    CLKOUT ^= 1; // Provide 0.5 Hz to Pin 1  
                // for checks  
    curs++;     // Update the current  
                // seconds counter  
    TMR1H = TMR1H_INI; // Reload counter high byte
```

```
    TMR1L = TMR1L_INI; // Reload counter low byte  
    PIR1.TMR1IF = 0; // Clear TMR1 Interrupt flag  
  }  
}
```

Dopo il setup hardware e l'inizializzazione delle variabili, il loop principale della seguente procedura "main" gestisce gli eventi secondo la logica di un Automa a Stati Finiti.

Lo stato corrente dell'automa è impostato nella variabile "Mode". Per prima cosa viene verificata la pressione sul pulsante SW1 per il reset del contatore di eventi, ed il riavvio del microcontrollore tramite l'istruzione assembler "reset".

Alla prima accensione con microcontrollore vergine, e solo in tale occasione, occorrerà premere il pulsante SW1 per impostare a zero il conteggio degli allarmi in EEPROM.

All'esecuzione del Setup, a seguito della pressione del pulsante, verrà anche forzato lo spegnimento del relè di comando dell'allarme, nel caso per qualsiasi motivo esso fosse rimasto attivo.

Gli eventi di attivazione dell'allarme vengono conteggiati nella variabile "AlmCnt".

Da notare che, nel caso in cui il ritorno della tensione di rete sia molto tardivo e si ecceda la capacità di supportare le operazioni fornita dal supercondensatore C5, per non perdere il conteggio degli allarmi si salva il valore di "AlmCnt" in EEPROM. La procedura di Setup ne prevede la rilettura al riavvio del microcontrollore:

```
void main() {  
  
  setup();          // Initialisations  
  
  while(1) {        // Forever Loop  
  
    if (Button(&PORTB,4,1,0)){ // Reset button pressed  
      STSLED = 1;  
      Delay_ms(500);  
      STSLED = 0;  
      AlmCnt = 0;  
      EEPROM_Write(CNTADDR, AlmCnt);  
      // Raz the Alarms counter in EEPROM  
      curms = 0; // Raz the elapsed ms for Blink cycle  
      BlkSts = 0; // Reset the Blink automa  
      {asm{reset}}; // Reset the MCU  
                  // (the program will restart)  
    }  
  }
```

Nel brano seguente vengono letti i valori analogici presenti sui pin connessi al cursore dei due trim-



Il nuovo stato è "Mode_TOL":

Se esso si porta al livello logico Zero, allora verrà

Nello stato "Mode_TOL" gestito dalla sezione che segue, si attende uno di due possibili eventi: il ritorno della tensione di rete prima dello scadere del Tempo di Tolleranza, per tornare nuovamente nello stato "Mode_SBY", oppure lo sfioramento di tale tempo. La condizione da soddisfare per quest'ultimo caso è leggermente complicata dalla gestione della modalità opzionale "BEEP" attivata tramite il primo dei due Dip Switch.

A seconda dello stato di questo interruttore, verrà selezionato il tempo limite "ttol" definito prima dalla lettura del trimmer RV1, oppure il tempo "MinTol" fissato dalle definizioni iniziali del codice, corrispondente a 2 secondi. Il superamento del tempo limite porta l'automa nello stato "Mode_ALM". La prosecuzione del codice è sempre vincolata all'interrupt di Timer-1, visto che si continua ad operare in modalità a basso consumo:

```
case Mode_TOL:    // Waiting ttol time before activating the alarm

    if (POWER == 1) {                // Mains is back before the alarm activation
        Mode = Mode_SBY;             // Just go back to the Stand By Mode
    } else {                          // Check end of Tolerance time
        if ( ((BeepMode==FALSE)&&(curs>=ttol))\
            ||((BeepMode==TRUE)&&(curs>=MinTol)) ) {
            SetAlm();                 // Start the alarm
            IncAlm();                 // Increment the alarms counter
            Mode = Mode_ALM;          // Enter in the Alarm Mode
        }
    }
    break;
```

Questa parte gestisce lo stato "Mode_ALM". Si attende il ritorno della tensione di rete, per resettare l'allarme e tornare in Stand-By uscendo definitivamente dalla modalità a basso consumo. Oppure si attende l'expiration del tempo massimo di allarme, scelto tra il valore "talm" impostato dalla lettura analogica del trimmer RV2, oppure dal valore "MinAlm" preimpostato ad 1 secondo all'inizio del programma. Si resta in tal caso in basso consumo, per gestire il ritorno della tensione di rete nello stato "Mode_BACK":

```
case Mode_ALM:    // Wait talm seconds before
                  // to stop the alarm
```

```
if (POWER == 1) { // Mains is back during Alarm
    ResetAlm();   // Stop the alarm
    if (BackMode==TRUE) Signal(); // Signal "Mains
                                // is back" if requested
    Mode = Mode_SBY; // Manage the Power Up event
} else {          // Check end of Alarm time

    if ( ((BeepMode==FALSE)&&(curs>=talm))\
        ||((BeepMode==TRUE)&&(curs>=MinAlm)) ) {
        ResetAlm();   // Suspend the alarm
        Mode = Mode_BACK; // Wait for the
                                // Power Up event
    }
}
break;
```

In "Mode_BACK", si attende il ritorno della tensione di rete e si segnala l'evento attivando l'allarme per due brevi impulsi consecutivi se previsto dall'interruttore "BACK" del Dip-Switch. Finché la rete non torna operativa, il PIC viene lasciato in modalità a basso consumo:

```
case Mode_BACK:    // Waiting for the Mains power coming back again

    if (POWER == 1) {                // Mains is back: exit to Stand By mode
        ResetAlm();                 // Force alarm off
        if (BackMode==TRUE) Signal(); // Signal "Mains is back" if requested
        Mode = Mode_SBY;
    }
    break;
```

Dopo l'automa principale ne è stato implementato un secondo a quattro stati, per la sola gestione del Led D5 connesso con RA0.

Questo Led ha lo scopo di mostrare il conteggio corrente degli eventi di caduta di tensione di durata superiore al Tempo di Tolleranza. Il contatore "AlmCnt" è resettabile premendo il pulsante SW1, gestito nelle prime linee di codice del ciclo principale, visto in precedenza. La gestione dei lampeggi tramite un automa è dettata dall'esigenza di non poter interrompere il ciclo principale con dei semplici "Delay", che impedirebbero ogni altra azione durante la loro esecuzione. I tempi di lampeggio sono definiti dalle costanti "CNTBLINK" impostata a 200 ms, e "CNTINTERV" che definisce la pausa tra due blocchi di lampeggi, impostata a 3 secondi (**Listato 1**).

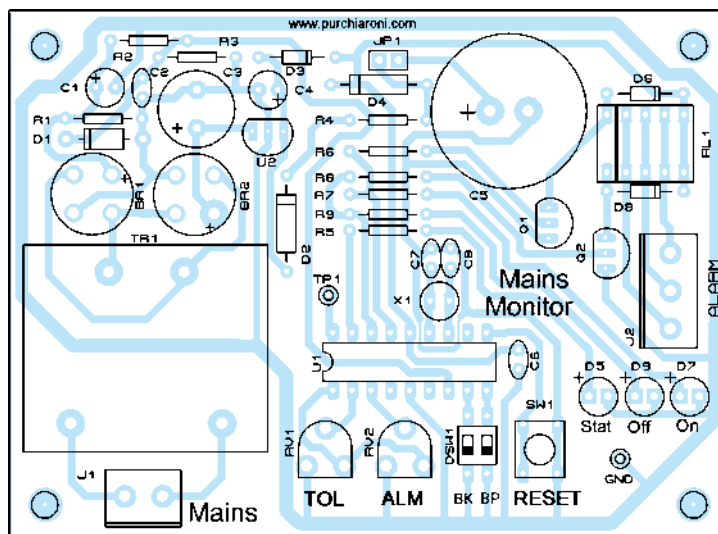


Elenco Componenti:

R1: 22 kohm
 R2: 1 kohm
 R3: 10 kohm
 R4: 22 ohm
 R5, R7, R9: 470 ohm
 R6, R8: 4,7 kohm
 RV1, RV2: Trimmer 22 kohm MO
 C1, C4: 47 μ F 16 VL elettrolitico
 C2, C6: 100 μ F ceramico
 C3: 1000 μ F 16 VL elettrolitico
 C5: 1,5 F 5,5 VL supercondensatore
 C7, C8: 22 pF ceramico
 D1: Zener 5,1 V / 500 mW
 D2, D4: SB140
 D3: 1N4007
 D6, D8: 1N4148
 D5: LED 3 mm Giallo
 D7: LED 3 mm Rosso
 D9: LED 3 mm Verde
 BR1, BR2: W04G
 Q1, Q2: BC547
 U1: PIC16F1827 (MF1723)
 U2: 78L05
 RL1: Relè 3V Doppia Bobina DPDT
 J1: Morsetto 2 poli passo 5 mm
 J2: Morsetto 3 poli passo 5 mm
 TR1: Trasformatore 230 Vac / 6 Vac 1,5 VA

Varie:

- Strip maschio 2 vie
- Jumper
- Circuito Stampato S1723 (95x70 mm)



CIRCUITI STAMPATI

Sia per la versione Analogica che per quella Digitale, sono stati progettati due circuiti stampati "Mono-faccia" e privi di ponticelli.

Questo faciliterà senz'altro la loro auto-realizzazione tramite i classici metodi di fotoincisione, o di trasferimento a caldo.

E' anche possibile mandare in stampa i PCB tramite il servizio di Futura Elettronica, disponibile all'indirizzo (ftlink.it/pcb).

Il PCB della versione Digitale è più grande per la presenza di un maggior numero di componenti elettronici. È possibile scaricare dalla pagina di presentazione di questo numero i file Gerber necessari per l'auto-costruzione.

REALIZZAZIONE

Nella versione analogica mostrata nel piano di montaggio, lo switch SW1 per la selezione della modalità di funzionamento è collegato con dei fili ed un connettore a sei poli al circuito stampato. Potete collegarvi uno switch a leva, oppure un qualsiasi commutatore a due vie.

La versione digitale del dispositivo, visibile nel piano di montaggio, prevede il montaggio su zoccolo del microcontrollore, due trimmer di tipo miniatura, un Dip-Switch a due vie ed un Jumper.

Il supercondensatore da 1.5F è un componente particolare, ma comunque reperibile.

In questa versione l'uscita verso l'allarme è fornita tramite un morsetto a tre vie, che permette di

Listato 1

```
// On power presence, signal the occurred alarms (one blink per each alarm)
if ((POWER == 1) && (AlmCnt > 0)) {
  switch (BlkSts) {
    case 0: // Initialize a new Blinks cycle
      n = AlmCnt;
      STSLED=1;
      curms = 0; // Start ms counting
      BlkSts = 1; // Go to "Wait to turn off the LED"
      break;
    case 1: // Wait CNTBLINK ms to turn off the LED
      if (curms >= CNTBLINK) {
        STSLED=0;
        n--; // Decrement residual Blinks counter
        curms = 0; // Start ms counting
        if (n > 0) BlkSts = 2; // Go to "Wait to turn on the LED"
        else BlkSts = 3; // Go to "Wait for next Blinks cycle"
      }
      break;
    case 2: // Wait CNTBLINK ms to turn on the LED
      if (curms >= CNTBLINK) {
        STSLED=1;
        curms = 0; // Start ms counting
        BlkSts = 1; // Go to "Wait to turn off the LED"
      }
      break;
    case 3: // Wait CNTINTERV for next Blinks cycle
      if (curms >= CNTINTERV) {
        curms = 0; // Start ms counting
        BlkSts = 0; // Initialize a new cycle
      }
      break;
  }
  if (BlkSts > 0) curms += TIC; // Increment the elapsed time counter (ms)
}

#ifdef LOWPOW
// If no external power is present, go in Low Power Mode
if (POWER == 0) {asm{sleep};} // Sleep. Awake on Timer1 interrupt.
#endif

Delay_ms(TIC); // Introduce a cycle delay
}
```

scegliere il tipo di attivazione NC o NO, a seconda dell'avvisatore disponibile.

Questo non è possibile nella versione analogica, per limiti circuitali.

La versione digitale offre quindi una maggiore versatilità e facilità d'uso rispetto alla versione analogica, grazie alla possibilità di configurare l'uscita dell'allarme secondo le proprie necessità.

Tuttavia, la versione analogica mantiene una semplicità strutturale che può essere vantaggiosa in alcune applicazioni specifiche.

La scelta tra le due versioni dipenderà quindi dalle esigenze specifiche dell'utente e dalle caratteristiche del sistema di allarme a cui il dispositivo verrà collegato.



 COSA OCCORRE ?		
Codice	Descrizione	Prezzo € IVA inclusa
PCBPRODUCTION	Servizio realizzazione PCB (il costo dipende dalle dimensioni del circuito stampato).	
SET1390COMP	Molti dei componenti utilizzati sono disponibili nel set di componenti assortiti.	17,00
IL MATERIALE VA RICHIESTO A:		
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA) Tel: 0331-799775 - www.futurashop.it		



L'ENERGIA PULITA che ti fa risparmiare!

Pannelli solari monocristallini semirigidi 12V



Sono composti da singoli e sottilissimi strati di cristallo di silicio che opportunamente trattati aumenta la loro conduzione. Le celle che compongono il pannello monocristallino sono facilmente riconoscibili grazie al loro colore molto scuro. La purezza del silicio che compone le celle monocristalline permette di ottenere la massima efficienza energetica e grazie a questa caratteristica è possibile installare un numero di pannelli inferiore rispetto ai tradizionali, permettendo così di occupare superfici meno estese. L'alta qualità del materiale utilizzato permette di avere una durata maggiore del pannello.

Pannelli solari monocristallini rigidi 12V



Regolatori di carica per pannelli fotovoltaici a 12 e 24V



Prezzi IVA inclusa.

SMART HOUSE CON ARDUINO

del **PROF. MICHELE MENNITI**



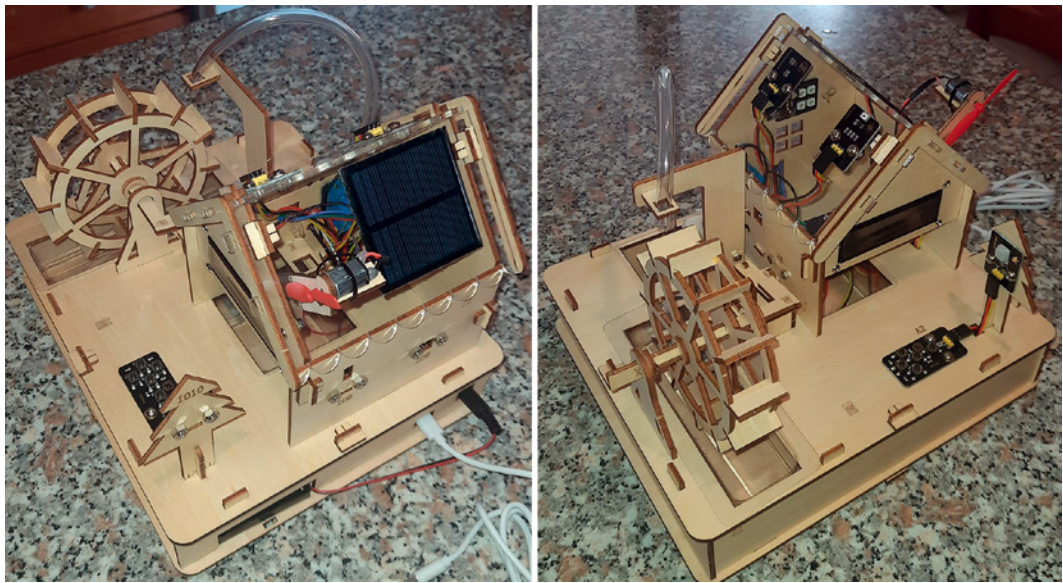
Realizziamo una Smart House automatizzata con sensori per luci, temperatura, umidità e altri parametri ambientali, gestiti da una scheda compatibile con Arduino UNO.

V

i proponiamo oggi un bellissimo kit progettato dalla Keyestudio, denominato "Smart Eco-Friendly House Kit" e reperibile sul sito www.futurashop.it col codice EDUKIT16. L'idea è quella di realizzare una villetta in miniatura completamente automatizzata, mediante l'utilizzo simultaneo di sensori e attuatori, che

possono agire in piena autonomia o essere gestiti mediante una tastiera ed un display LCD; il cuore del progetto è una board compatibile con Arduino UNO, con eccellenti caratteristiche, denominata KS0486 Keyestudio Plus. Il produttore del kit ha realizzato una pagina WIKI ricca di contenuti indispensabili per far funzionare tutto.

Fig. 1
La Smart House
Arduino montata.



Il kit viene fornito in una scatola con l'immagine della villetta completa funzionante; per avere un'idea del risultato finale riportiamo la foto di quella montata da noi (**Fig. 1**).

Una volta aperta la confezione (**Fig. 2**) troviamo sei pannelli in balsa (il tipico legno da modellismo) contenenti tutte le parti necessarie per la realizzazione della struttura, già intagliate, che si possono

staccare con una leggera pressione sui pezzi stessi; la parte che costituirà il tetto trasparente è invece realizzata su un pannello in plexiglass, anch'esso già intagliato; la meccanica è completata da viti, dadi, distanziali, tubi, vaschette, fascette, un'elica in plastica, cavetti jumper per i collegamenti, e alcuni piccoli attrezzi, indispensabili per il montaggio.

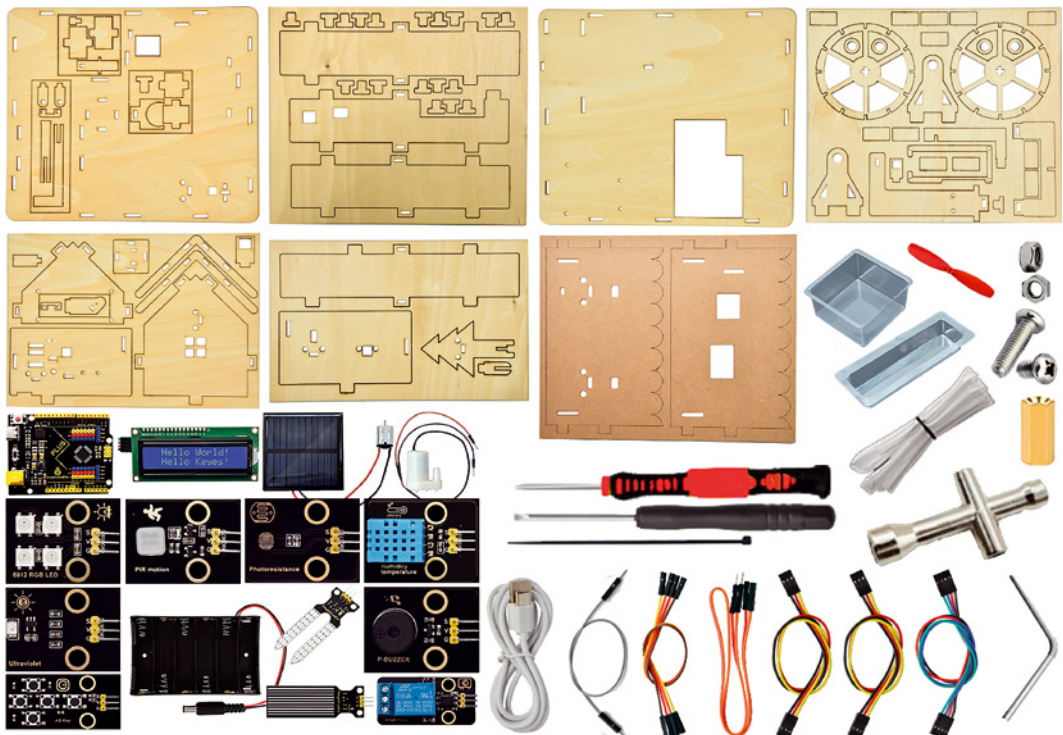


Fig. 2
La confezione del
kit Smart House
EDUKIT16.

L'elettronica è composta da:

- Board KS0486 Keystudio Plus, perfettamente compatibile con Arduino UNO;
- Display LCD 16x2 con interfaccia I²C;
- Modulo RGB;
- Modulo PIR;
- Fotoresistore;
- Sensore di temperatura e umidità XHT11;
- Pulsantiera con cinque tastini;
- Sensore dell'umidità del suolo;
- Sensore di livello dell'acqua;
- Modulo relè 5V;
- Buzzer passivo;
- Sensore raggi UV;
- Pompa elettrica per l'acqua;
- Pannello solare con motorino DC;
- Portabatterie 6 x AA.

La board e tutti i vari moduli e sensori sono di ottima qualità, facilmente identificabili dalle serigrafie, oltre che mediante le numerose foto presenti nella WIKI. Il montaggio richiede tempo e la giusta attenzione, sulla WIKI sono presenti sia un video che una pagina con tutte le fasi del montaggio ma, considerata la nostra esperienza diretta, abbiamo deciso di preparare un manuale in italiano ricco di immagini, con la descrizione di ogni singolo passaggio.

In questo articolo di presentazione ci dedichiamo interamente alla parte software e a tal fine riporteremo i collegamenti dei singoli moduli alla board, in modo da poterli testare tutti facilmente prima di montarli definitivamente nella struttura.

Nel manuale in italiano fornito all'interno del kit trovate i dettagli per il montaggio e tutte le necessarie indicazioni su come utilizzare la Smart House. Abbiamo descritto tutti gli sketch, approfondendo alcuni elementi utili per una migliore comprensione dell'uso dei moduli. Infine, abbiamo predisposto alcuni sketch aggiuntivi, scaricabili dalla sezione download del nostro sito.

Tutti gli sketch messi a disposizione dal produttore sono scaricabili dalla WIKI, sezione Arduino Tutorial: <https://docs.keyestudio.com/projects/KS0576/en/latest/docs/1.Arduino%20Tutorial/Arduino%20Tutorial.html>.

Da questa pagina si accede tramite menu alla sezione **Software Installation**, dalla quale è possibile scaricare tutto il necessario per i test circuitali:

1. IDE 2.x: questo download va eseguito solo se non si è ancora installata l'IDE di Arduino, tenendo conto comunque che anche se la WIKI evidenzia l'utilizzo della versione 2.1.0, noi abbiamo testato tutto con successo anche con la versione 1.8.19.

2. Driver della board: contrariamente all'Arduino UNO originale, questa board usa un convertitore USB-Seriale basato sul chip CP2102; la prima operazione da fare è quindi quella di caricare nel PC i driver di questo convertitore, in caso contrario la board non potrà essere configurata correttamente e l'IDE non la riconoscerà. Nella sezione **Download driver** occorre cercare la board KS0486 Keystudio Plus e quindi selezionare il driver per Windows o Mac, seguendo poi le chiare istruzioni presenti sulla pagina che si aprirà. I nostri test sono stati effettuati in ambiente Windows.
3. Librerie: servono tre librerie aggiuntive a quelle standard presenti nell'IDE, si possono scaricare dal link proposto, col nome **Libraries.zip**, per poi installarle seguendo le istruzioni.
4. Al punto successivo della stessa pagina, **Download Code**, bisogna cliccare sul link proposto per scaricare il file **Arduino_Code.zip**, che contiene tutti gli sketch per testare i vari moduli utilizzati nel kit (**Fig. 3**).

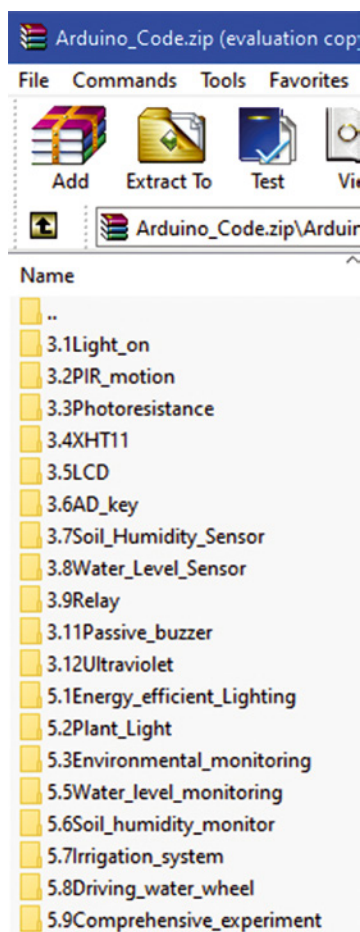
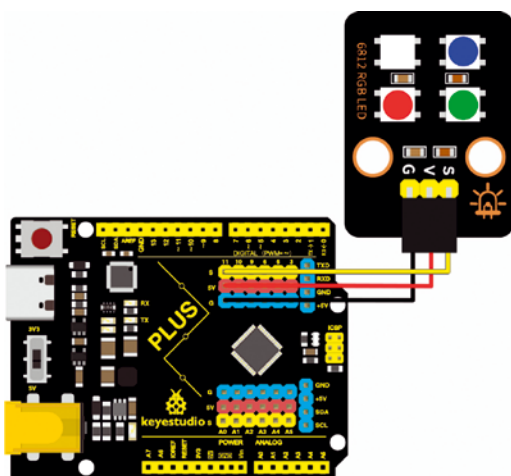


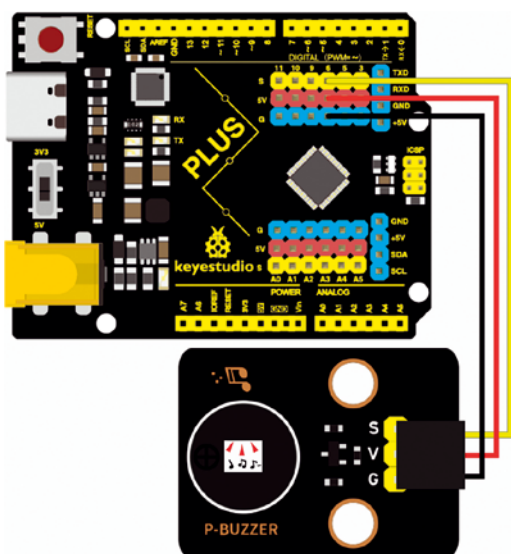
Fig. 3
Gli sketch
per la Smart
House.

→ **Fig. 4**
Collegamenti mo-
dulo RGB.



Apriamo l'IDE e selezioniamo la board Arduino UNO e la COM alla quale essa è collegata, quindi possiamo iniziare i nostri test, caricando gli sketch uno per volta e controllando che tutto funzioni correttamente. Naturalmente, nel caso in cui non si riscontri l'effetto descritto, occorrerà controllare prima di tutto i collegamenti e poi eventuali condizioni fisiche e ambientali della struttura, in base al tipo di sensore o attuttore sotto test; per esempio il sistema composto dal pannello solare che alimenta un motorino con elica non funzionerà se testato in casa, in quanto è indispensabile l'energia solare o almeno una fonte di luce molto potente. Vediamo ora alcuni esempi del codice fornito.

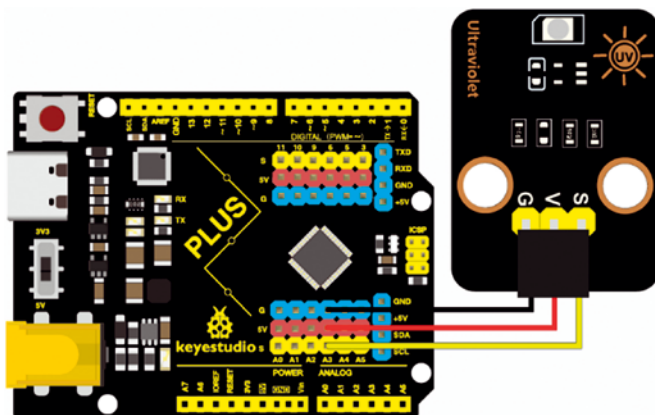
→ **Fig. 5**
Collegamenti
buzzer passivo.



3.1Light_on: test del modulo RGB montato all'interno della villetta e collegato al pin digitale 11 (**Fig. 4**); questo sketch richiede la libreria Adafruit Neopixel, una delle tre messe a disposizione dalla WIKI, ma conviene installarla direttamente dal gestore delle librerie dell'IDE, essendo presente una versione più recente. L'effetto è quello dell'accensione fissa dei quattro LED del modulo RGB con questi colori: bianco, blu, rosso e verde. Al fine di approfondire le funzionalità di questo modulo abbiamo predisposto uno sketch alternativo che presentiamo nel manuale d'uso.

3.11Passive_buzzer: test del buzzer passivo montato all'interno della villetta e collegato al pin digitale 6 (**Fig. 5**); è uno sketch molto semplice nel cui loop il pin 6 viene posto prima su High e poi su Low, con un intervallo di 500 us; in tal modo il buzzer genera una nota fissa da 1 kHz; variando leggermente uno dei due intervalli è possibile variare anche il tono della nota. Nel manuale d'uso abbiamo inserito uno sketch alternativo che permette di ottenere una nota variabile.

→ **Fig. 6**
Collegamenti
e test sensore UV
al chiuso.



```
COM3
The Photocurrent value : 0mV
UV Index = 0
The Photocurrent value : 0mV
UV Index = 0
The Photocurrent value : 0mV
UV Index = 0
The Photocurrent value : 0mV
UV Index = 0
The Photocurrent value : 0mV
UV Index = 0
```



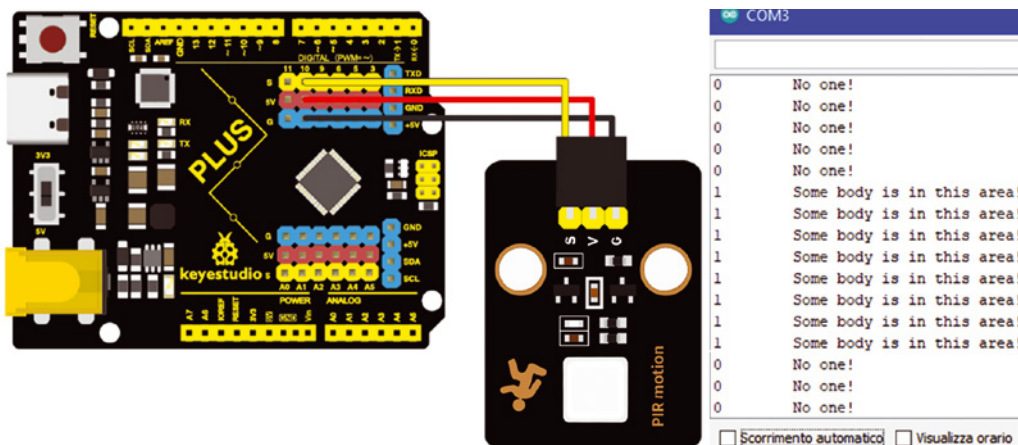


Fig. 7
Collegamenti
e test sensore
PIR.

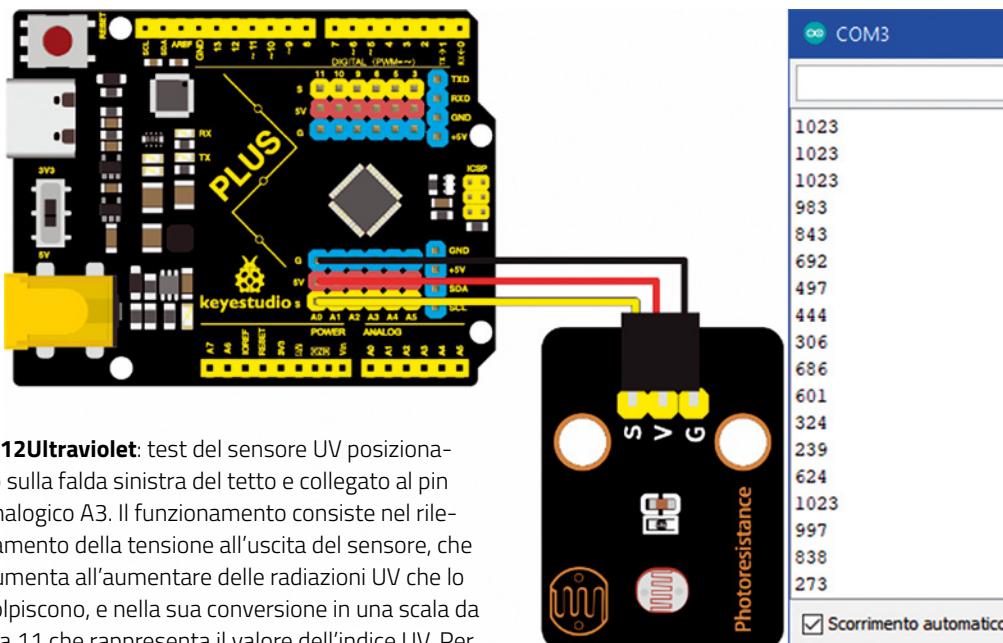


Fig. 8
Collegamenti
e test fotore-
sistore.

3.12Ultraviolet: test del sensore UV posizionato sulla falda sinistra del tetto e collegato al pin analogico A3. Il funzionamento consiste nel rilevamento della tensione all'uscita del sensore, che aumenta all'aumentare delle radiazioni UV che lo colpiscono, e nella sua conversione in una scala da 0 a 11 che rappresenta il valore dell'indice UV. Per osservare i risultati occorre aprire il Serial Monitor dell'IDE, naturalmente il test deve essere eseguito esponendo il tetto della villetta alla luce solare, altrimenti si otterrà sempre risultato $uv=0$, come nel nostro caso (**Fig. 6**).

3.2PIR_motion: test del sensore PIR posizionato sul supporto a forma di alberello nel cortile e collegato al pin digitale 10. Questo sensore rileva la presenza delle radiazioni infrarosse emesse dal corpo umano o da un animale di dimensioni idonee. In mancanza di radiazioni il LED rosso presente sul sensore è acceso, il pin "S" del sensore si trova al livello logico 0 (Low) e sul Serial Monitor appare in sequenza il messaggio "0 No one!" (Nessuno); è sufficiente passare una mano davanti al sensore per ottenere l'immediato cambio di stato: il LED

rosso si spegne, il pin "S" del sensore passa al livello logico 1 (High) e sul Serial Monitor esce in sequenza il messaggio "1 Some body is in this area!" (C'è qualcuno in quest'area!). Togliendo la mano per qualche istante resta l'indicazione di allarme, poi il LED si riaccende e ritorna il messaggio iniziale, fino a nuovo cambio di stato (**Fig. 7**).

3.3Photoresistance: test del fotoreistore posizionato sulla falda sinistra del tetto (vicino al sensore UV) e collegato al pin analogico A0. Questo sensore rileva le variazioni della luminosità ambientale, il semplice firmware di test invia al Serial Monitor il valore della lettura analogica, che è 1023 in condizioni di massima luminosità per diminuire man mano che l'ambiente si fa più buio.



Per vedere il funzionamento in tempo reale, una volta caricato il programma e aperto il Serial Monitor basta prendere un qualsiasi oggetto sufficientemente largo e avvicinarlo al fotoresistore fino a toccarlo, si potranno osservare le variazioni del valore della lettura analogica (**Fig. 8**).

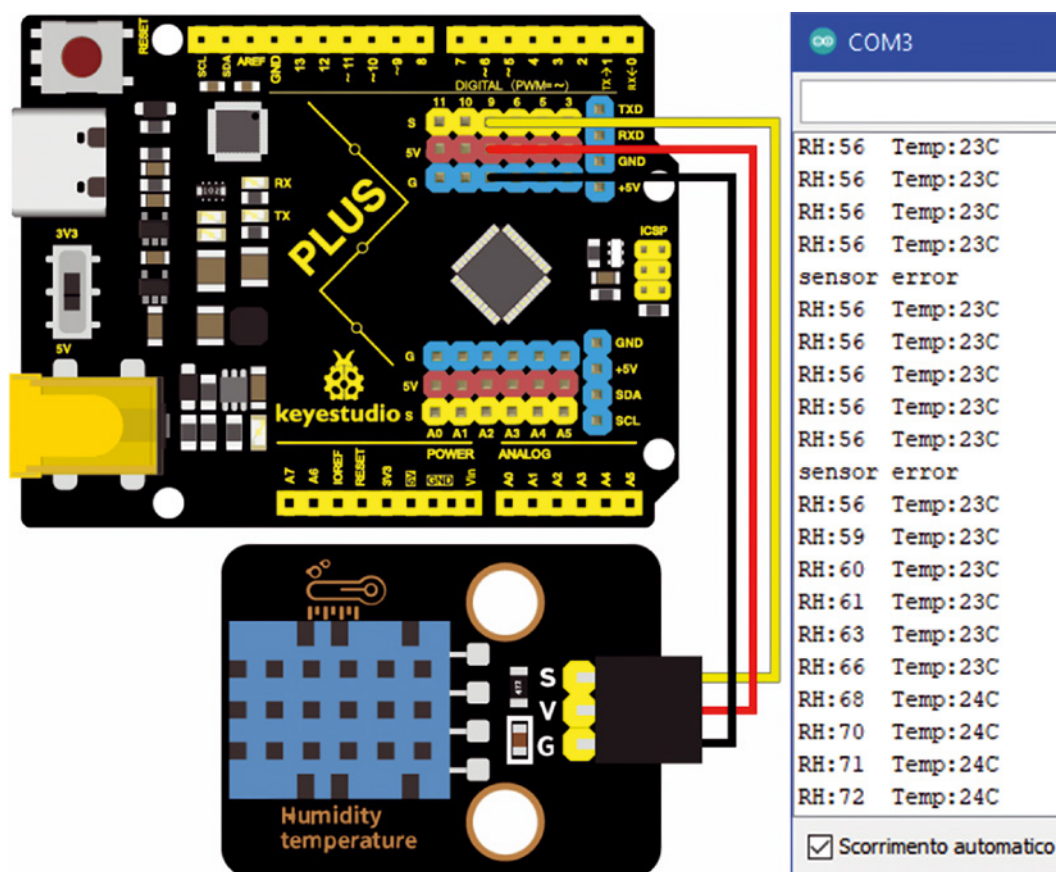
3.4XHT11: test del sensore di umidità e temperatura presente all'interno della villetta, sulla stessa parete del buzzer passivo e collegato al pin digitale 9. Questo sketch richiede la libreria specifica xht11, che è una delle tre installate in precedenza. Il programma crea un array di 32 bit per la lettura dei dati provenienti dal sensore e una volta caricato è immediatamente operativo, inviando al Serial Monitor i dati relativi all'umidità ed alla temperatura all'interno della villetta. Per osservare il comportamento del sensore basta riscaldare l'ambiente, magari con un asciugacapelli tenuto a debita distanza e si potrà osservare subito l'aumento della temperatura, mentre per l'umidità servirà del vapore; per esempio si potrebbe mettere dell'acqua nella vaschetta interna alla villetta e dirigere verso di essa il flusso di aria calda

dell'asciugacapelli, facendo però attenzione a non farla fuoriuscire. Tra una lettura e la successiva c'è una pausa di un secondo.

Nel modello da noi testato ogni tanto compariva un errore di lettura del sensore, probabilmente dovuto a qualche disturbo presente nel nostro laboratorio, ma le misure sono comunque risultate precise (**Fig. 9**); peraltro è un particolare facilmente eliminabile modificando lo sketch in modo da saltare semplicemente la visualizzazione in caso di errore di lettura e così vengono mostrati solo i valori corretti di lettura.

3.5LCD: test del display LCD I²C localizzato sul frontespizio della villetta e collegato ai pin specifici presenti sulla board (**Fig. 10**); facilmente la libreria richiesta, la classica LiquidCrystal_I2C, sarà già presente nell'elenco, in caso contrario si potrà utilizzare quella scaricata con le altre due dalla WIKI. All'inizio dello sketch c'è la dichiarazione dell'indirizzo esadecimale e del numero di caratteri e righe del display (0x27, 16, 2); una volta caricato nella board si attiverà la retroilluminazione e il display mostrerà due frasi di esempio "Hello World!"

Fig. 9
Collegamenti e test
sensore umidità e
temperatura.



e "Hello Keyes!". Se la leggibilità dovesse risultare scarsa occorrerà posizionarsi con gli occhi in linea retta rispetto al display e si avrà il miglior contrasto possibile.

3.6AD_Key: test della pulsantiera localizzata nel cortile di fronte l'alberello e collegata al pin analogico A2. La tecnica adottata per identificare diversi pulsanti collegati allo stesso pin analogico l'abbiamo usata in altre occasioni sui numeri della nostra rivista. Si tratta di realizzare un partitore resistivo multiplo mediante un certo numero di resistori in serie; in base al pulsante premuto il partitore fornisce al pin analogico una tensione differente, lo sketch deve solo identificare il range nel quale essa ricade. I valori dichiarati dalla WIKI sono molto precisi quindi il test potrà essere considerato corretto se i risultati saranno molti simili a quelli visibili in **Fig. 11**, con SW1=1023, SW2=815, SW3=614, SW4=406 e SW5=207.

Nel caso si dovesse riscontrare un malfunzionamento, del tipo che premendo SW2 la risposta sarà sempre 1023-SW1 o più in generale risposte molto diverse da quelle attese, probabilmente si tratterà del deviatore presente sulla board KS0486 Keystudio Plus settato su 3V3 anziché su 5V; comunque questa cosa è chiaramente spiegata nel manuale di montaggio che forniremo a corredo del kit.

3.7Soil_Humidity_Sensor: test del sensore di umidità del suolo, posizionato all'interno della viletta, sulla parete opposta a quella del buzzer

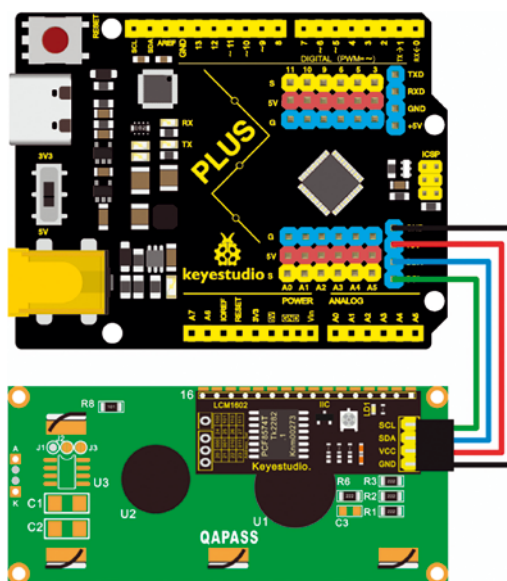


Fig. 10
Collegamenti display LCD.

passivo e che finisce all'interno della vaschetta. L'idea è quella di riempire la vaschetta con del terreno nel quale si può mettere una piantina, quando il terreno si sta asciugando il sensore permette di rilevare la scarsa umidità e di dare il via alla pompa elettrica per prelevare l'acqua dalla vaschetta esterna adibita a serbatoio e irrigare il terreno; comunque in questo sketch di esempio viene solo rilevato il tasso di umidità, senza attivazione del relè. Il sensore è collegato al pin analogico A6; una curiosità a proposito di questo pin: sulla board Arduino UNO originale non è presente in quanto basata sull'ATmega328P-PU in formato pth (a foro

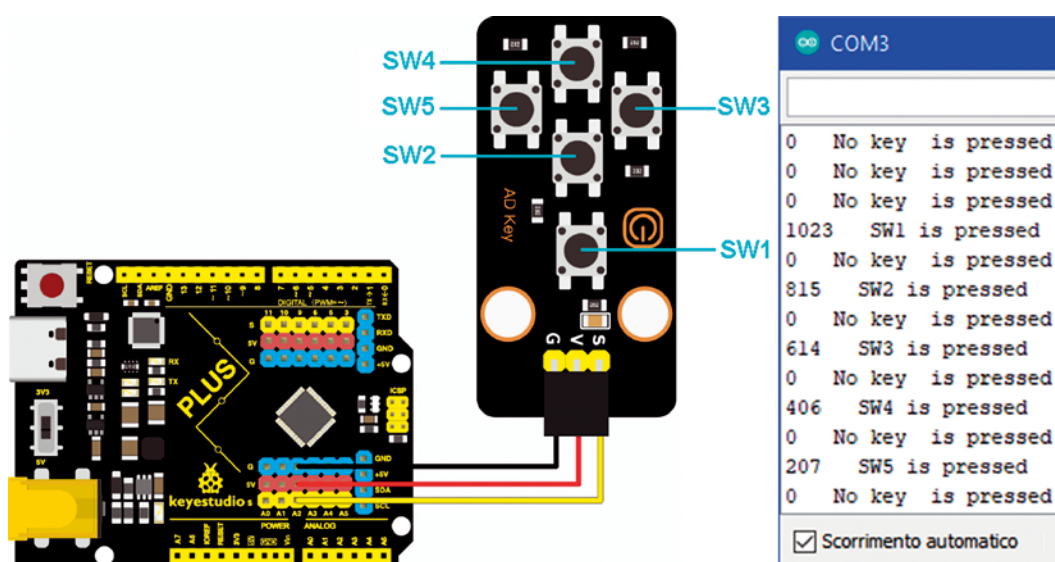
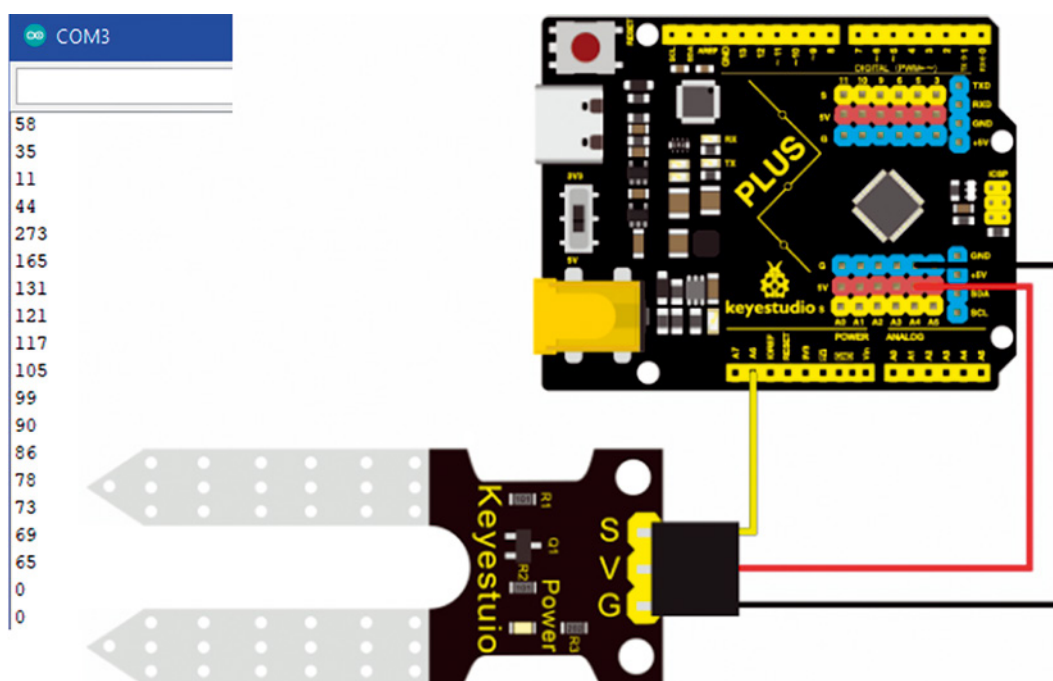


Fig. 11
Collegamenti e test pulsantiera analogica.

→ **Fig. 12**
Collegamenti e test
sensore umidità del
suolo.



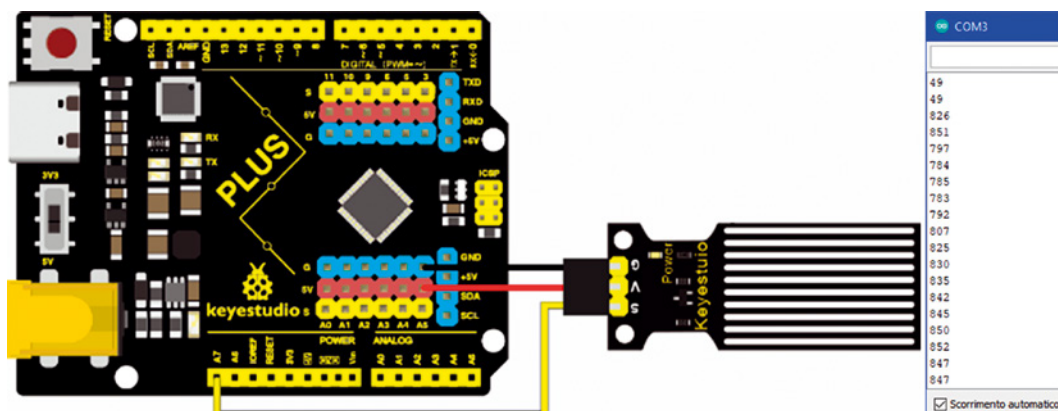
passante), mentre sulla board compatibile adottata dal produttore di questo kit è stato montato l'ATmega328P-AU, che è la versione smd e che è dotata di due pin analogici in più, A6 e A7. Lo sketch di test mostra il tasso di umidità del terreno che riempie la vaschetta interna alla viletta. Per controllare il buon funzionamento del sensore è sufficiente toccare le due piastrelle con un dito bagnato e si potrà osservare la variazione della lettura, il valore aumenta all'aumentare dell'umidità (Fig. 12).

3.8Water_Level_Sensor: test del sensore di livello dell'acqua, posizionato all'esterno della viletta, nella vaschetta contenente la pompa elettrica, e

collegato al pin analogico A7. Questo sensore, nella nostra viletta smart, controlla che l'acqua sia presente nel serbatoio costituito dalla vaschetta, per evitare l'azionamento della pompa a vuoto. La verifica si può fare velocemente con una siringa riempita di acqua: una volta caricato lo sketch e aperto il Serial Monitor si può iniziare a versare lentamente l'acqua nella vaschetta e si vedrà il valore di lettura aumentare proporzionalmente all'aumento del livello dell'acqua (Fig. 13).

3.9Relay: test del relè posizionato all'interno della viletta, sulla stessa parete del sensore di umidità del suolo e collegato al pin digitale 5 (Fig. 14). Anche in questo caso serve il Serial Monitor, ma

→ **Fig. 13**
Collegamenti e test
sensore di livello
dell'acqua.



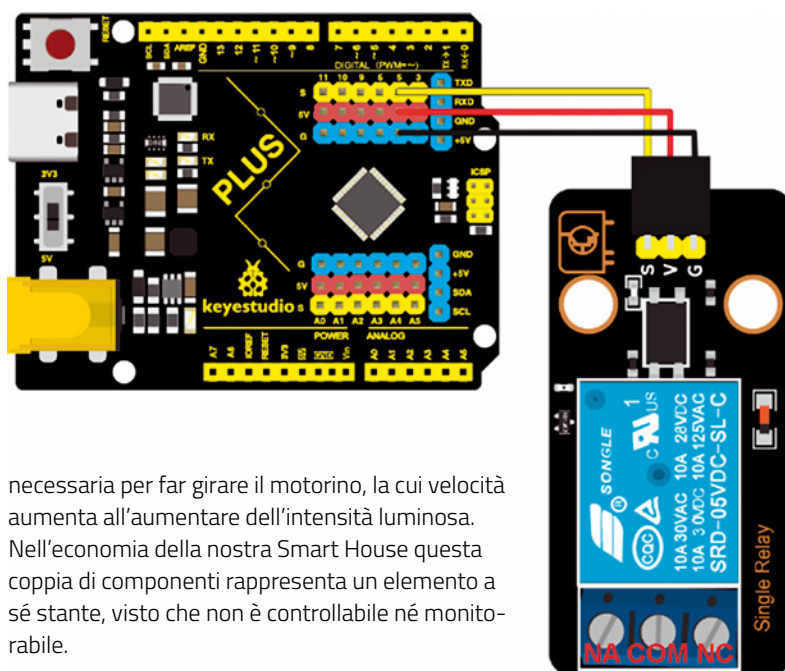
solo per inviare un comando e non per ricevere dati. Il comando consiste nell'immettere la lettera "d" nella riga di comando del Serial Monitor e subito dopo premere INVIO, si sentirà istantaneamente lo scatto del relè che, dopo circa mezzo secondo, tornerà in posizione di riposo. Considerando l'assorbimento del relè, questo test richiede che la board sia alimentata mediante il pacco batterie esterno. Da qui in avanti gli sketch sono esempi operativi che controllano piccoli gruppi di sensori e attuatori. Qui ci limiteremo ad elencarne le caratteristiche essenziali, il loro funzionamento sarà approfondito nel manuale d'uso che forniremo a corredo del kit, che conterrà anche le immagini con i collegamenti.

5.1Energy_efficient_lighting: questo sketch monitora contemporaneamente il fotoreistore e il sensore PIR e all'occorrenza pilota il modulo RGB. Lo sketch simula la segnalazione di una presenza all'esterno della villetta in condizioni di luce molto scarsa, infatti se si oscura il fotoreistore in modo da scendere sotto il valore 200 e si attiva il sensore PIR passandogli davanti una mano, il modulo RGB accende in sequenza tutti e quattro i suoi LED con luce bianca e sul Serial Monitor il LED diventa "on".

5.2Plant_light: questo sketch utilizza il modulo della pulsantiera e quello RGB; il principio su cui si basa è quello del miglioramento del processo di crescita delle piante fornendo loro luce a diverse lunghezze d'onda. I pulsanti coinvolti in questo sketch sono SW3, SW4 e SW5, alla prima pressione di un pulsante i LED si accendono con il colore associato al pulsante premuto, alla successiva si spengono.

5.3Environmental_monitoring: serve per il monitoraggio ambientale e utilizza questi moduli: fotoreistore, ultravioletto, umidità e pressione e display LCD. Vengono attivati i tre sensori, che misurano l'intensità della luce esterna, la sua componente di raggi UV e la temperatura e umidità interne della villetta, i risultati vengono visualizzati tutti contemporaneamente sul display LCD.

5.4Application_of_Solar_Energy: questo test, pur seguendo la numerazione, non è accompagnato da uno sketch, infatti non c'è alcun controllo software in quanto i due elementi coinvolti non sono interfacciati con la board; si tratta di un pannello solare che alimenta direttamente un motorino sul quale è innestata un'elica; in presenza di luce solare sufficiente il pannello eroga la tensione



necessaria per far girare il motorino, la cui velocità aumenta all'aumentare dell'intensità luminosa. Nell'economia della nostra Smart House questa coppia di componenti rappresenta un elemento a sé stante, visto che non è controllabile né monitorabile.

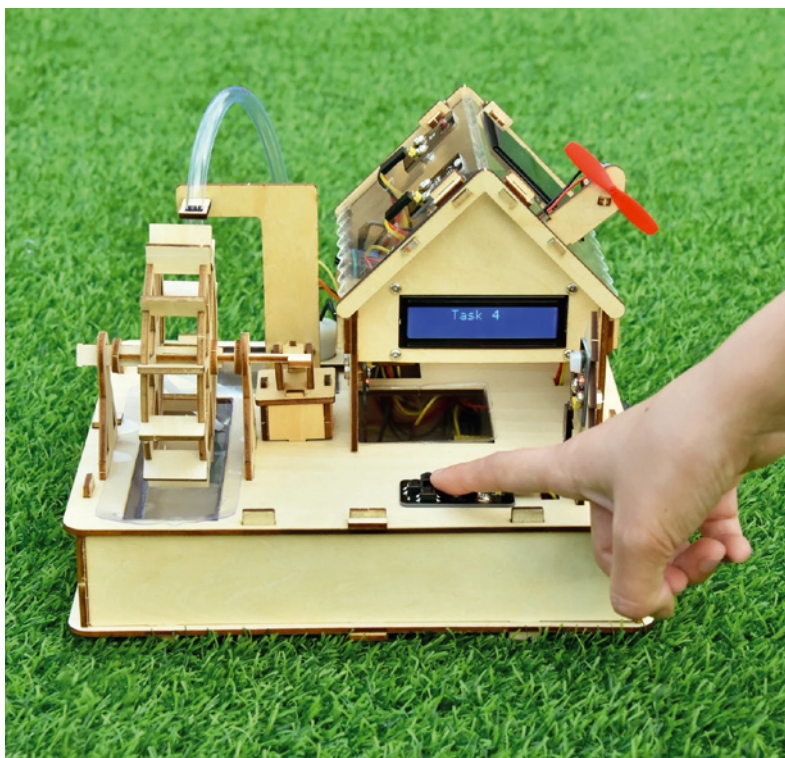
5.5Water_Level_Monitoring: serve per il monitoraggio del livello di acqua presente nella vaschetta serbatoio nella quale è immersa la pompa elettrica e utilizza questi moduli: sensore di livello dell'acqua, buzzer, display LCD. Il livello dell'acqua viene visualizzato costantemente sul display; la presenza di acqua con un livello tra 50 e 500 viene considerata come serbatoio rifornito; se l'acqua scende ad un livello inferiore a 50 il buzzer viene attivato ed emette un suono bitonale con frequenze di 532 Hz e 639 Hz, simulando una sirena di allarme. Se invece l'acqua sale ad un livello eccessivo, tale da superare il valore 500, si attiva una sirena bitonale con frequenze di 1000 Hz e 1500 Hz.

5.6Soil_Moisture_Monitoring: serve per il monitoraggio dell'umidità del terreno situato nella vaschetta all'interno della villetta e utilizza questi moduli: sensore dell'umidità del suolo, buzzer, display LCD.

Il funzionamento è praticamente identico a quello dello sketch precedente: se l'umidità scende ad un livello minore o uguale a 20 il buzzer viene attivato ed emette un suono bitonale con frequenze di 532 Hz e 639 Hz, simulando una sirena di allarme. Se invece l'umidità sale ad un livello eccessivo, tale da raggiungere almeno il valore 200 sul display, si attiva una sirena bitonale con frequenze di 1000 Hz e 1500 Hz.

5.7Irrigation_System: questa configurazione simula un impianto di irrigazione automatico ed

Fig. 14
Collegamenti del
modulo relè.



 La Smart House Arduino in funzione.

utilizza questi moduli: sensore dell'umidità del suolo, buzzer, relè, sensore del livello dell'acqua, display LCD, pompa elettrica. Lo sketch tiene costantemente sotto controllo sia il livello dell'acqua nella vaschetta serbatoio, situata all'esterno della villetta, nella quale è immersa la pompa elettrica, sia l'umidità del terreno contenuto nella vaschetta interna alla villetta e visualizza questi valori sul display LCD. Quando l'umidità del terreno scende sotto la soglia impostata via software, se il serbatoio contiene acqua a sufficienza (anche questa soglia è preimpostata) viene attivato il relè e parte la pompa elettrica che irrignerà il terreno, fino al superamento della soglia di umidità minima.

5.8 Water_Wheel_System: questo test utilizza la pulsantiera analogica e il relè; è un'interessante dimostrazione del principio dei vasi comunicanti di Stevino, in particolare si tratta del fenomeno conosciuto come sifonamento, che è il principio fisico secondo il quale "un liquido contenuto in due o più contenitori comunicanti tra loro, in presenza di gravità, raggiunge lo stesso livello originando un'unica superficie equipotenziale". Nella pratica della nostra Smart House abbiamo la vaschetta serbatoio localizzata più in basso rispetto a quella della ruota idraulica; prelevando l'acqua del serbatoio tramite la pompa elettrica ed immettendola nella vaschetta della ruota idraulica, si creerà un sistema di circolazione continua, in quanto l'acqua tornerà nella vaschetta serbatoio tramite il tubo pieghevole in dotazione.

5.9 Comprehensive_experiment: come si deduce dal nome questo sketch gestisce tutti i moduli del kit. Esso è costituito da quattro degli esperimenti (task) visti in precedenza più uno nuovo, selezionabili mediante la pressione del tasto SW1 della pulsantiera. Per i nostri lettori abbiamo predisposto una variante di alcuni degli sketch del produttore, al fine di approfondire la comprensione del funzionamento dei relativi moduli; tra questi anche una nostra versione dello sketch 5.9, con alcune interessanti implementazioni e modifiche. Tutti gli sketch saranno illustrati nel manuale d'uso e resi disponibili nella sezione download del nostro sito.

La "Smart Eco-Friendly House Kit" rappresenta un'opportunità straordinaria per immergersi nel mondo dell'automazione domestica e dell'elettronica. Questo progetto non solo consente di costruire una casa in miniatura completamente automatizzata, ma offre anche un'ampia gamma di esperienze educative attraverso l'uso di sensori e attuatori. Il manuale in italiano fornito nel kit è ricco di dettagli sia per il montaggio che per l'utilizzo della Smart House, con spiegazioni chiare e approfondite di tutti gli sketch. Inoltre, la disponibilità di sketch aggiuntivi scaricabili dal sito consente di espandere ulteriormente le funzionalità della casa automatizzata, permettendo agli utenti di personalizzare e adattare il sistema alle proprie esigenze. Questo kit è ideale per studenti, hobbisti e appassionati di elettronica che desiderano sviluppare le loro competenze in un contesto pratico e stimolante.



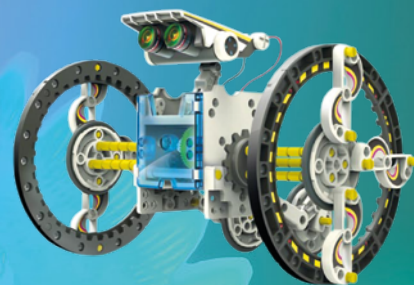
COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
EDUKIT16	Smart Eco-Friendly House Kit con manuale	89,00

IL MATERIALE VA RICHIESTO A:

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

SERVOMOTORI, INDISPENSABILI PER GLI APPASSIONATI DI ROBOTICA!



INGRANAGGI IN METALLO

cod. MC946R

€ 8,50



SERVO ANALOGICO AD ALTE PRESTAZIONI CON ROTAZIONE DI 180°

- Coppia di torsione:
10,5kg/cm (4,8V),
13kg/cm (6V)
- Velocità a 4,8V:
0,20sec/60° senza
carico, velocità a 6V:
0,17sec/60° senza
carico
- Alimentazione:
da 4,8 a 7,2V
- Dimensioni:
40,7x19,7x42,9mm
- Peso: 55g

cod. SERVO213

€ 11,00



SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione:
13kg/cm (4,8V) - 15kg/
cm (6V)
- Velocità a 4,8V:
0,20sec/60° senza
carico, velocità a 6V:
0,16sec/60° senza
carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
40,6x39,5x20mm
- Peso: 55g

cod. MC996R

€ 9,00



SERVO ANALOGICO CON ROTAZIONE DI 360°

- Coppia di torsione:
9,4kg/cm (4,8V) - 11kg/
cm (6V)
- Velocità a 4,8V: 0,19sec
/60° senza carico,
velocità a 6V: 0,15sec
/60° senza carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
42,7x40,9x20mm
- Peso: 55g

cod. SERVO860M9C

€ 9,00



MINI SERVO ANALOGICO CON ROTAZIONE CONTINUA DI 360°

- Coppia di torsione:
1,8kg/cm (4,8V)
- Rapporto di riduzione:
320:1
- Velocità a 4,8V: 0,12
sec/60° senza carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
22,6x32x12mm
- Peso: 12g

cod. SERVO09METAL

€ 8,00



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione:
1,8kg/cm (4,8V)
- Rapporto di riduzione:
320:1
- Velocità a 4,8V: 0,12
sec/60° senza carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
22,6x11,8x26,7mm
- Peso: 12g

INGRANAGGI IN PLASTICA

cod. SERVO6001HB

€ 20,00



SERVO MOTORE CON ROTAZIONE DI 180°

- Coppia di torsione:
6,7 kg·cm (a 6 Vdc)
5,8 kg·cm (a 4,8 Vdc)
- Velocità a 4,8V:
0,16 sec senza carico
Velocità a 6V:
0,14 sec senza carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
40,7x20,5x39,5mm
- Peso: 43g

cod. SERVO860LC

€ 9,90



SERVO ANALOGICO CON ROTAZIONE CONTINUA DI 360°

- Coppia di torsione:
5,5kg/cm (4,8V)
- Velocità a 4,8V:
0,22sec/60° senza
carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
40x39,1x20mm
- Peso: 38g

cod. SERVO209

€ 8,00



SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione:
5,5kg/cm (4,8V) -
6,5kg/cm (6V)
- Velocità a 4,8V:
0,19sec/60° senza
carico, velocità a 6V:
0,15sec/60° senza
carico
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
40,6x39,5x20mm
- Peso: 42,5g

cod. SERVO207

€ 9,90



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione:
2,3kg/cm
- Velocità: 0,18 sec/60°
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
28,5x13,3x30,6mm
- Peso: 17g

cod. SERVO206

€ 7,50



MINI SERVO ANALOGICO CON ROTAZIONE DI 180°

- Coppia di torsione
1,2kg/cm (4,8V) - 1,5kg/
cm (6V)
- Velocità a 4,8V:
0,12sec/60°, velocità a
6V: 0,11sec/60°
- Alimentazione:
da 4,8 a 6V
- Dimensioni:
22,5x12x29mm
- Peso: 9g

ANCHE I DETTAGLI PIÙ PICCOLI DIVENTANO VISIBILI!

**MICROSCOPI DIGITALI IN GRADO DI CATTURARE IMMAGINI
CON UN'INGRANDIMENTO FINO A 1200X.**

MICROSCOPIO DIGITALE 1-1200X CON MONITOR LCD A COLORI DA 7"

Microscopio digitale con ampio monitor LCD a colori da 7 pollici e ingrandimento fino a 1200X. Dispone di una telecamera a colori che riprende l'oggetto da ingrandire e lo visualizza in alta risoluzione sullo schermo LCD.

Le foto e i video degli oggetti ingranditi possono essere salvati su una SD card inserita nell'apposito slot (max 128GB, acquistabile separatamente). Il microscopio dispone di una base stabile in alluminio e di doppia luce a LED con bracci flessibili. La distanza del punto di osservazione del campione può essere facilmente regolata tramite l'asta a cremagliera che consente di ottenere una distanza che va da 10 a 130 mm. Lo schermo è ruotabile di 360° e inclinabile. Può funzionare tramite batteria al litio integrata oppure tramite l'alimentatore USB in dotazione.



cod. LCDMICROS7

€154,00



Prezzi IVA inclusa.

MICROSCOPIO DIGITALE CON INGRANDIMENTO FINO A 600X CON MONITOR LCD DA 4,3"

Microscopio digitale portatile, dotato di telecamera a colori con sensore da 3,6 Megapixel, monitor LCD a colori da 4,3", illuminatore composto da 8 potenti LED a luce bianca, batteria al litio ricaricabile integrata, slot per micro SD card (max 64GB) e staffa a ventosa.



cod. LCDMICROS

€102,00

MICROSCOPIO USB DIGITALE CON INGRANDIMENTO FINO A 500X

Microscopio USB digitale a colori da 2 Megapixel dotato di stand con supporto regolabile in altezza. Si collega alla porta USB e consente di catturare e visualizzare immagini e video direttamente sullo schermo del computer. Dispone di 8 LED bianchi, con intensità luminosa regolabile.

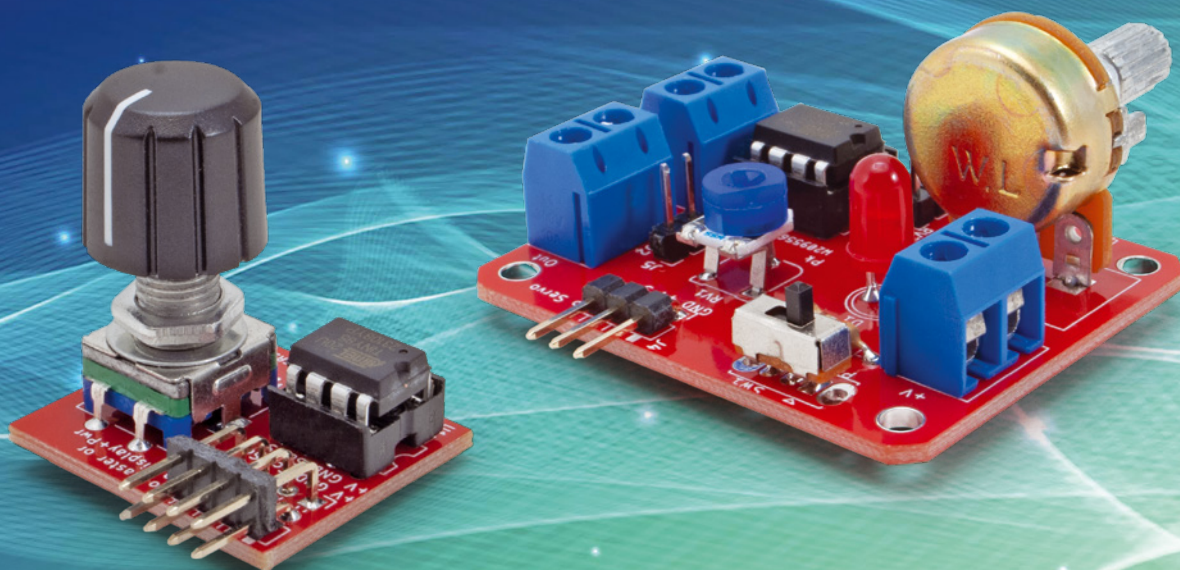
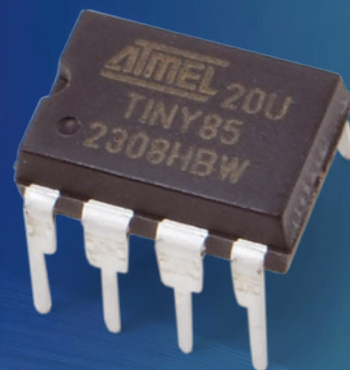


cod. CP745

€28,00

BREAKOUT CON ATTINY85

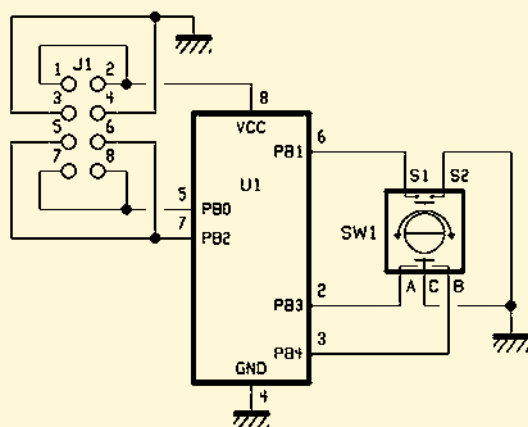
di DANIELE DENARO



Scopri il potenziale del microcontrollore ATtiny85 in applicazioni pratiche come la gestione di encoder rotativi e PID, sfruttando l'efficiente collegamento I2C.

N

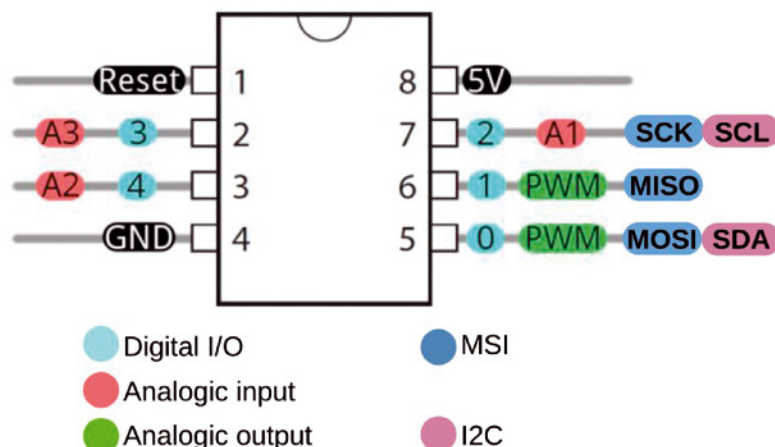
el numero 276 della rivista (Dicembre 23 - Gennaio 24) è stato presentato il piccolo ed economico ATtiny85 da utilizzare come versatile componente "multipurpose" in progetti elettronici e circuiti. Oltre a far parte di progetti più complessi, ATtiny, può anche essere usato come principale componente in configurazioni minime come lo sono, per esempio, i moduli "breakout". È questo l'argomento dell'articolo, in cui vedremo la realizzazione di due moduli di comodo utilizzo. Per chi si fosse perso l'articolo precedente, rivediamo velocemente le caratteristiche di ATtiny85. È un processore della serie Atmel (ora Microchip) AVR, quindi della stessa serie del processore ATmega328 che supporta la scheda Arduino Uno.



È questo il motivo per cui è relativamente semplice utilizzare l'ambiente di sviluppo Arduino IDE anche per ATtiny85, pur con qualche limitazione dovuta all'hardware.

ATtiny85 è piccolo **Fig. 1**, versatile ed economico. Per poterlo utilizzare con l'ambiente di sviluppo di Arduino, dovete selezionare la scheda hardware dedicata a lui. In questo articolo faremo riferimento all'ambiente "AttinyCore" che potete scaricare da

Fig. 1
ATtiny85
e pinout.



http://drazzy.com/package_drazzy.com_index.json. Una volta caricata la scheda (tramite il gestore di schede), utilizzate la "Board Attiny25/45/85" e quindi fate l'uploading dello sketch tramite la voce "Upload Using Programmer" del menù "Sketch". Ma, la prima volta che usate un ATtiny85 vergine, dovete prima scaricare il "Bootloader" con il comando "Bourn Bootloader" del menù "Tools". C'è ancora una **fondamentale** differenza con l'uso classico di IDE per Arduino: **non potete scaricare gli sketch direttamente!** Infatti ATtiny non ha USB. Per questo motivo dovete utilizzare un Arduino Uno come "programmer". Ovvero un Arduino su cui è stato caricato lo sketch "ArduinoISP" presente negli esempi. Arduino diventa così un programmatore in grado di programmare i nostri ATtiny. In **Fig. 2** potete vedere lo schema di collegamento per questo scopo. Mentre in **Fig. 3** è schematizzata la procedura di selezione del modello. Se il processo vi sembra complicato, vi consigliamo di recuperare l'articolo precedente in cui questo processo era spiegato con maggiori dettagli. Inoltre in quell'articolo è stata anche illustrata la costruzione di un programmatore con funzionalità di sperimentatore su basetta "breadboard". Costruire questo programmatore può risultare molto pratico nel caso abbiate intenzioni di utilizzare ATtiny più volte. Il modello di compilazione "AttinyCore" è più recente rispetto a quello originale di Damellis e permette fra l'altro l'utilizzo di I2C (Wire) che nel precedente modello non era implementato, per questo utilizzeremo "AttinyCore" nelle applicazioni descritte in questo articolo. I due moduli "breakout" che proporremo in questo articolo sono la gestione di un encoder rotativo ed un controllore PID **Fig. 4**. In ambedue i casi l'alimentazione prevista è 4.4-5 V.

ENCODER ROTATIVO

L'encoder rotativo meccanico, sempre più utilizzato nelle apparecchiature elettroniche, è utile per migliorare la sensibilità di un potenziometro senza ricorrere ad un potenziometro multi-giro. L'encoder rotativo è un componente di natura digitale, a differenza di un potenziometro, che è una resistenza variabile e quindi un componente di natura analogica. Infatti il risultato della sua rotazione è un valore numerico e corrisponde al numero di passi di rotazione, fatti in un verso o nell'altro. La rotazione è continua, ovvero senza fine, e serve ad incrementare o a decrementare un contatore. Da questa descrizione, risulta chiaro che l'encoder rotativo va gestito in modo digitale; ovvero con

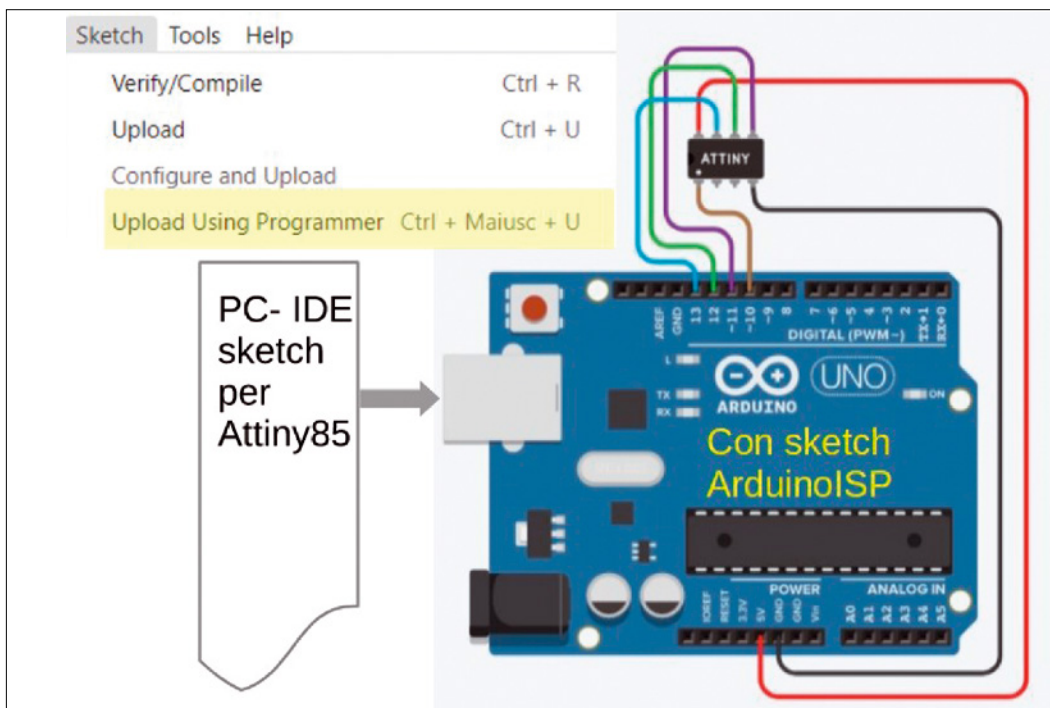


Fig. 2
Program-
mazione di
ATtiny85,
collegamenti
hardware.

Fig. 3
Programmazione
di ATtiny85, uso
dell'IDE di Arduino.

un algoritmo contatore di impulsi. Infatti l'encoder rotativo meccanico corrisponde ad una coppia di interruttori attivati in sequenza. Per discriminare il verso di rotazione (incremento o decremento), si usa un protocollo di impulsi detto a quadratura. Ovvero, durante la rotazione, vengono generati due impulsi quadrati sfasati tra loro di 90 gradi. Chiamando A e B le due sequenze di impulsi, A risulta in anticipo rispetto a B, ruotando in un verso, e in ritardo, ruotando in verso contrario **Fig. 5**. Come si vede, anche dalla figura, è necessario rilevare il cambiamento di stato (fronte di discesa o di salita). Naturalmente può essere fatto tranquillamente con Arduino, ed infatti esiste una libreria ("Encoder") già pronta per questo. Questa libreria può utilizzare le funzionalità di "interrupt" dei processori ATM oppure no. È chiaro che con l'utilizzo degli interrupt l'efficienza è maggiore. Purtroppo gli interrupt di ATtiny85 non sono così sofisticati come quelli dei processori più grandi. Per utilizzare in modo efficiente la gestione dell'encoder con ATtiny85 è meglio utilizzare un software ad hoc per questo processore. In rete esiste un suggerimento efficiente che verrà utilizzato per questa implementazione (cfr. Technoblogy - Bounce-Free Rotary Encoder o Technoblogy - Simple Rotary Encoder Interface [Updated]). In sostanza viene implementata una funzione di interrupt con anche la gestione del possibile

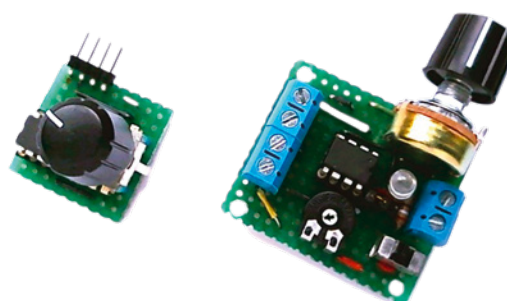
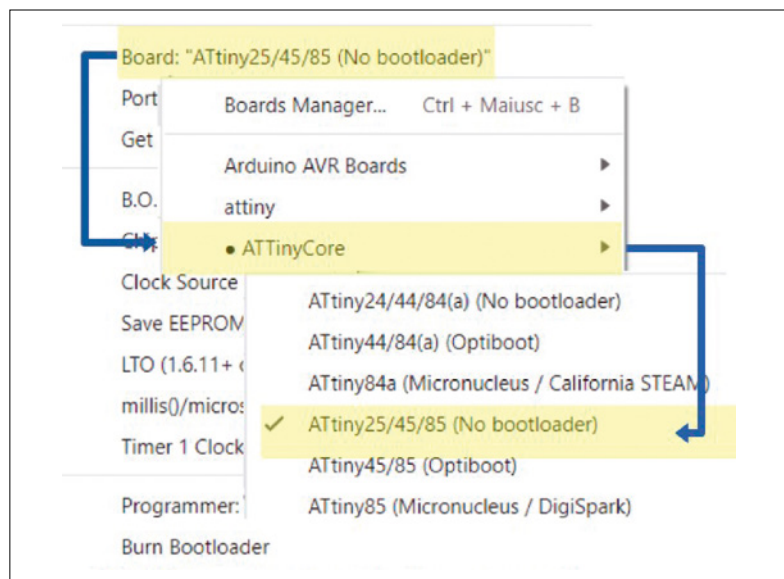
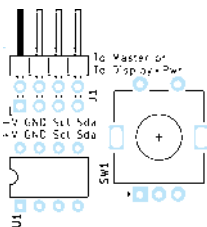


Fig. 4
Breakout:
Server
encoder, PID.



Elenco Componenti:

SW1: Encoder Rotativo
U1: ATtiny85-20P
(MF1746)

Varie:

- Strip maschio 2x4 vie 90°
- Zoccolo 4+4
- Circuito Stampato S1746 (29x25 mm)

rimbalzo (*bounce*), mediante la memorizzazione degli stati (Fig. 6). La gestione degli interrupt su ATtiny85 è meno sofisticata che su Arduino e si attiva solo sul cambio dello stato e non sul livello

(alto o basso) raggiunto. Per questo motivo il conteggio è doppio.

Perché utilizzare ATtiny85 per gestire un encoder rotativo? Ci sono due possibili motivi:

- Realizzare un server da collegare con link I2C ad un sistema come Arduino per scaricarlo dell'incarico di gestire gli impulsi dell'encoder. Con il collegamento I2C è possibile indirizzare più di un encoder in modo da avere la semplice lettura di più contatori, senza caricare il sistema, e con soli due fili impegnati.
- Realizzare un contagiri con display per rotazioni lente. Per esempio per il conteggio di spire di avvolgimento. Ci stiamo riferendo ad encoder rotativi con contatti meccanici e con debole scatto alla posizione, per cui non è pensabile l'utilizzo per rotazioni veloci, come con dei motori.

Per questa applicazione useremo l'encoder a bassissimo costo di FuturaElettronica disponibile con il codice ENCODER12MM. Questo encoder ha 20 passi per rotazione (18° per passo) ed un ulteriore pulsante indipendente realizzato spingendo sul perno. Sia per l'Encoder Server sia per il contagiri useremo lo stesso circuito con diverso software. Nel primo caso useremo il collegamento I2C da parte di uno "slave". Cosicché Arduino agirà da master e richiederà valori ad ATtiny85 mediante una libreria predisposta. Nel secondo caso, invece, ATtiny85 agirà da master ed utilizzerà, come slave, un display numerico a 8 caratteri.

ENCODER HARDWARE

Per l'hardware è stato disegnato un piccolissimo circuito con due soli componenti: l'ATtiny85 e l'encoder (Fig. 7). È stato realizzato con una "Stripboard" (millefori con collegamenti a strisce). Ma per chi preferisse un PCB (ad uno strato), dal sito della rivista è possibile scaricare il file zip contenente i file Gerber (file di maschere e dettagli costruttivi). Come visibile dallo schema elettrico il circuito è semplicissimo. Il doppio connettore J1 è utile quando lo si usa come contagiri collegato al display per permettere l'ingresso e l'uscita di alimentazione. Infatti quando è collegato come server ad un computer bastano quattro pin.

SOFTWARE PER L'ENCODER SERVER

Il software per ATtiny85 non fa altro che gestire l'encoder mediante interrupt e nel contempo leggere eventuali comandi di un byte che arrivano dalla linea I2C. La gestione dell'encoder è asincrona, ovvero la funzione che rileva lo scatto ed incrementa o decrementa il contatore, viene chiamata

Fig. 5
Encoder
rotativo
meccanico.

Fig. 6
Routine di gestione
encoder
meccanico.

```
Configurazione:
pinMode(EncoderA, INPUT_PULLUP);
pinMode(EncoderB, INPUT_PULLUP);
PCMSK = 1<<EncoderA; // Configure pin change interrupt on A
GIMSK = 1<<PCIE; // Enable interrupt
GIFR = 1<<PCIF; // Clear interrupt flag

Funzione eseguita all'interrupt:
ISR (PCINT0_vect) {
  int a = PINB>>EncoderA & 1;
  int b = PINB>>EncoderB & 1;
  if (a != a0) { // A changed
    a0 = a;
    if (b != b0) {
      b0 = b;
      changeValue(a == b);
    }
  }
}

Funzione conteggio:
void changeValue (bool up) {
  Count = (up?1:-1);
```

da un cambiamento sul pin dell'encoder. Invece la rilevazione di un eventuale comando (I2C) è lasciata al classico loop di Arduino (**Fig. 8**). Benché sia sempre possibile modificare e ricaricare il software su ATtiny85, per cambiare il suo indirizzo o per configurarlo come server o come contagiri, si è preferito gestire da software questi cambiamenti di configurazione. In questo modo si evita di estrarlo e riprogrammarlo ogni volta. Il problema è stato quello di evitare micro-switch, sia perché i pin di input sono esauriti, sia per le dimensioni veramente ridotte del PCB. Per il cambiamento di indirizzo (configurato inizialmente come 55) è previsto un comando dell'utilizzatore (master). Inoltre nel caso in cui, per qualche motivo, non si conosca l'indirizzo modificato di un server, è possibile farlo dire a lui con un truccetto. Poiché non può essere interrogato dal master (che deve conoscerlo per interrogarlo!), è prevista una fase iniziale allo startup dove il server si collega (come master) ad un indirizzo fittizio (di valore 1) spedendogli il suo indirizzo. In questo modo se l'utilizzatore si pone inizialmente come slave (con indirizzo 1), può ricevere il dato. Infine il passaggio da server a contagiri e viceversa è attivato dalla pressione prolungata del pulsante dell'encoder (circa 4 secondi). Pertanto, oltre a fornire il valore del contatore o lo stato del pulsante, il server esegue anche le seguenti operazioni.

Verifica alla partenza la EEPROM :

- Se il primo byte ha un valore compreso tra 2 e 127 utilizza questo valore come suo indirizzo piuttosto che quello definito nel programma, nel caso agisca come slave (server encoder)..

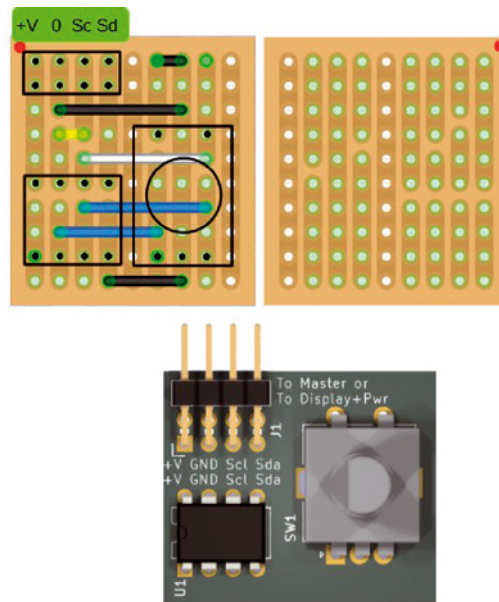


Fig. 7
Stripboard
fronte / retro
e rendering
del PCB.

- Se il byte 10 ha valore 1 allora commuta in modalità contagiri: si pone come master I2C e pilota il display TM1650 con il numero dei giri (40 impulsi per ogni giro) e nient'altro.

Se in modalità server:

- Dopo la verifica della EEPROM si pone dapprima in modalità master per spedire ad un indirizzo di comodo (indirizzo 1) il suo proprio indirizzo (come slave) e subito dopo nella sua effettiva modalità slave (server). In questo modo un sistema che si colleghi come slave (con indirizzo 1) può ricevere l'indirizzo dell'encoder.

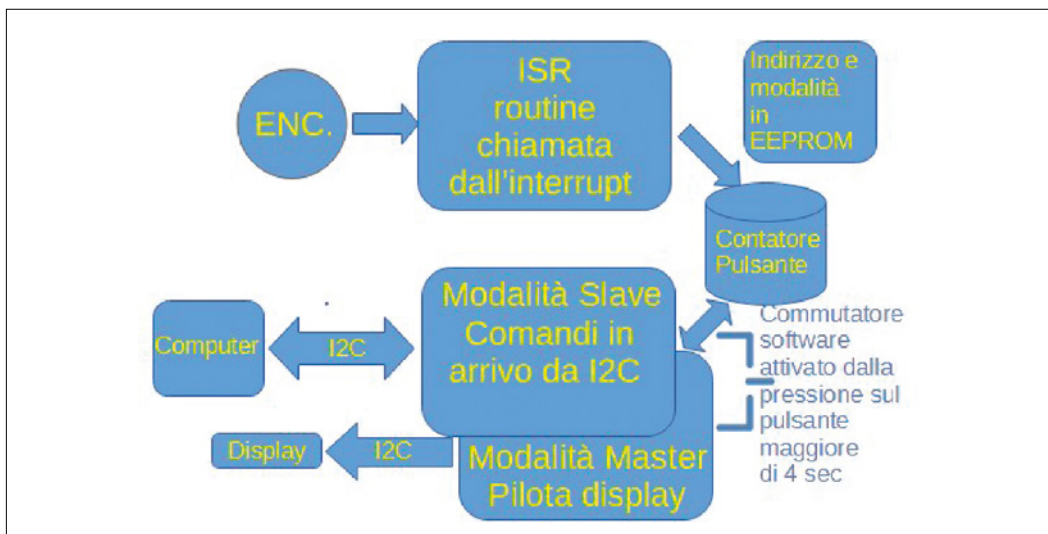


Fig. 8
Struttura
del software
su ATtiny85
come server
o come con-
tagiri.

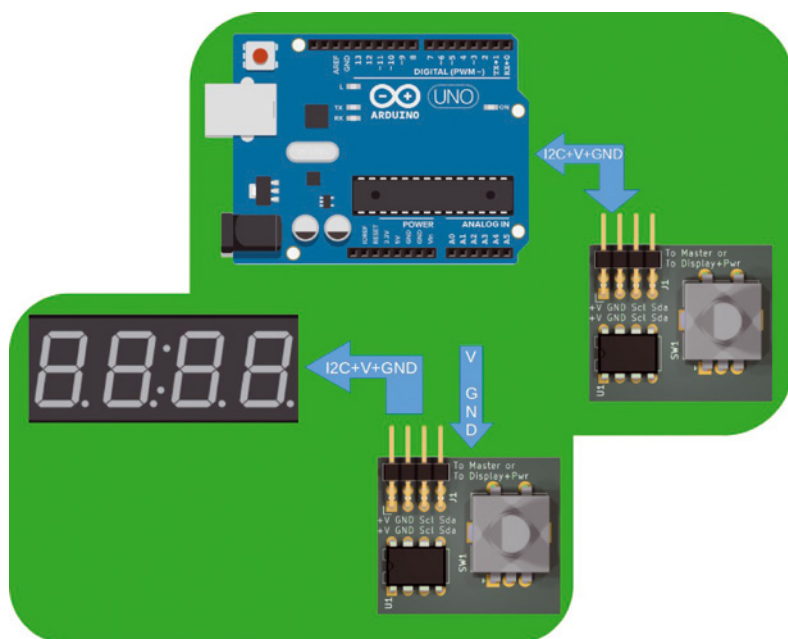


Fig.9
Server Encoder
o Contagiri.

→ La scheda Enco-
der montata.



- Può ricevere il comando di modifica del suo indirizzo (byte comando + byte nuovo indirizzo). In questo modo è possibile modificare on-line l'indirizzo del "device". L'indirizzo verrà salvato nella EEPROM.

A causa della gestione interrupt di ATtiny85, il conteggio dei passi è doppio. Infatti l'interrupt rileva la modifica di stato, ovvero sia la salita che la discesa dei livelli di tensione.

Sul sito della rivista è scaricabile il software per ATtiny85 con il nome **"TinyEncoderServer.ino"** e la libreria per un display a 7 segmenti con 4 cifre e interfaccia I2C con TM1650 **"TM1650.h"**.

Per l'utilizzo da parte di un sistema come Arduino o altro, è stata preparata una libreria C++ con la classe EncServer da istanziare e con le funzionalità riportate in **Tabella 1**.

Ovviamente se l'encoder cambia il suo indirizzo, bisognerà indirizzarlo, successivamente, con il nuovo valore.

Fuori dalla classe "EncServer" è presente inoltre la funzione GetDevAddress(). Questa funzione si pone come slave di indirizzo 1 e attende la ricezione dell'indirizzo spedito dal server.

Il server deve essere quindi attivato contestualmente, infatti spedisce il suo indirizzo solo alla partenza.

Questa gestione degli indirizzi si è resa necessaria per evitare di fare l'uploading del programma su ATtiny85 ogni qual volta bisognasse cambiare l'indirizzo configurato (che è 55), per esempio nel caso di più server encoder sulla stessa linea.

La libreria per la gestione del server encoder più uno "sketch" Arduino (TinyEncServerLib.ino) con una completa prova delle funzionalità è presente nel sito della rivista con i nomi **"TinyEncServerLib.h"** e **"ArduinoTestTinyserverLib.ino"**.

TABELLA DELLE FUNZIONI DELLA LIBRERIA DI GESTIONE DEL SERVER ENCODER (metodi della classe EncServer)

begin()	si collega come "master" (prima funzione da chiamare)
testDevice(indirizzo)	ritorna falso se la connessione al "device" di questo indirizzo dà errore
getEncoder(indirizzo)	ritorna il valore del contatore dell'encoder
getButton(indirizzo)	ritorna lo stato del pulsante (0/1)
setNewAddress(indirizzo, nuovo indirizzo)	chiede al "device" di cambiare il suo indirizzo
restartCounter(indirizzo)	chiede al "device" di azzerare il suo contatore

Tabella 1
Funzioni della
libreria del server
encoder.

SOFTWARE PER CONTAGIRI

Per collegare l'encoder ad un display I2C come per esempio il display a 7 segmenti a 4 cifre di Futura Elettronica (codice 2846-4DIGMODI2C), bisogna collegarlo, questa volta come master ed utilizzare una libreria di gestione "TM1650.h" del controller I2C del display. In questo modo si tratterà semplicemente di inviare il conteggio del numero di giri al display, lasciando al pulsante il compito di azzerare il conteggio. Un giro sarà compiuto in 20 passi (ma con conteggio 40). La gestione dell'encoder è invece invariata. Come detto il cambiamento di funzione è fatto in modo automatico dal software in seguito alla pressione prolungata del pulsante e la modalità è registrata in EEPROM.

CONTROLLORE PID

Ricordiamo brevemente che un controllore PID è un sistema che permette di gestire un attuatore in modo più preciso e stabile possibile, "controllando" l'azione. L'uscita dell'attuatore viene confrontata con il riferimento d'ingresso.

La sua "devianza", cioè l'errore viene elaborato per comandare l'attuatore (**Fig. 10**).

L'acronimo PID significa Proporzionale-Integrativo-Derivativo. Infatti l'errore, tra il riferimento e l'azione effettiva, viene moltiplicato per un fattore proporzionale, sommato ad un fattore del suo integrale ed infine sommato ad un fattore della sua derivata. Il risultato serve a comandare l'attuatore.

Il controllore PID è ampiamente utilizzato in ambito industriale per la sua semplicità e versatilità, anche se ha sempre bisogno di una messa a punto contestuale. Infatti può produrre oscillazioni o al contrario essere troppo lento.

In ambito Arduino esistono già diverse librerie PID, per cui abbiamo preso spunto da queste, in particolare dalla libreria di "Brettbeauregard" (PID library), cercando di configurarla per ATtiny85. Per chi volesse una interessante esposizione delle problematiche affrontate dalla libreria, può fare riferimento alla pagina:

<http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>

In sostanza il controllore ATtiny85 PID riceve un

valore di riferimento (setpoint) tramite una lettura analogica su un potenziometro (normalizzata tra 0 e 1). Questo valore viene confrontato con un valore analogico letto su un ingresso di feedback (normalizzato tra 0 ed 1) e la differenza viene elaborata dai calcoli PID. Il risultato attiva una uscita PWM. In ogni caso per permettere di "bypassare" il controllo PID ed attivare direttamente l'uscita PWM, è previsto uno switch (in realtà un "jumper") per collegare la lettura del potenziometro al valore PWM. In quest'ultima modalità, ATtiny85 si comporta semplicemente come un regolatore PWM (e l'ingresso di feedback viene ignorato).

Collegare il ponticello significa attivare il controllo PID. Quindi, senza ponticello il modulo si comporta

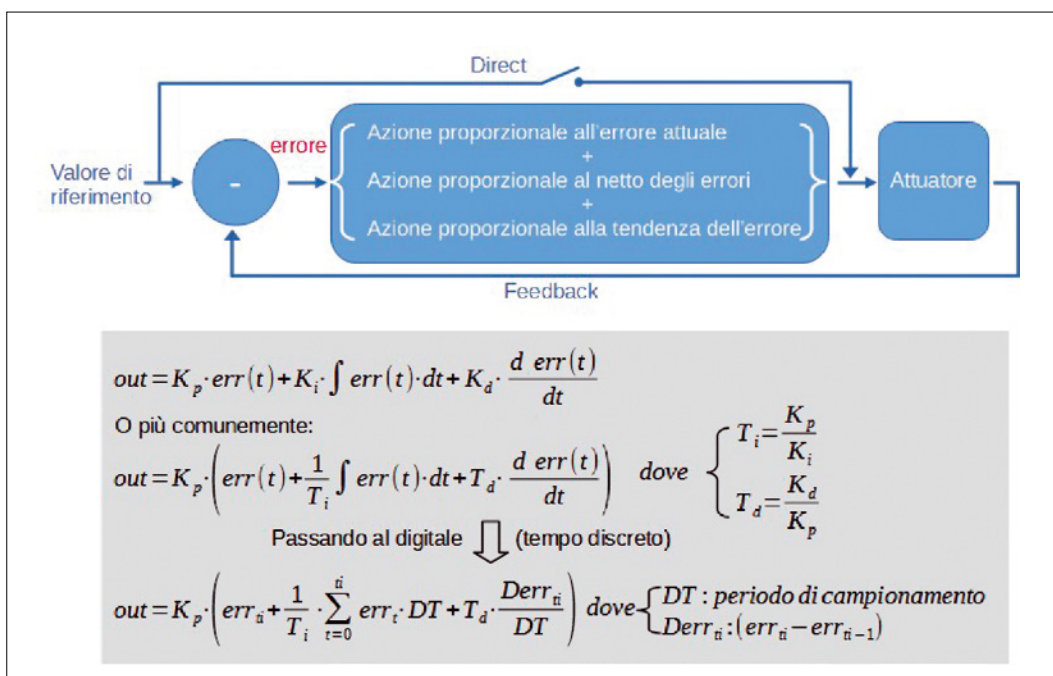
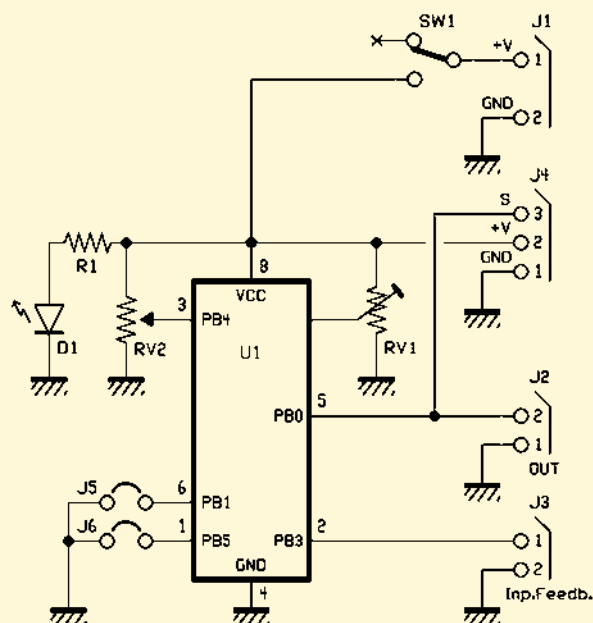


Fig. 10
Funzionamento
di un regolatore
PID.



originariamente come driver PWM o driver Servo. Infatti come vedremo più avanti è possibile attivare in sequenza tre modalità di azione: driver PWM, PWM con riferimento a 0.5, driver Servo-attuatore. È possibile regolare, tramite un trimmer, il solo coefficiente proporzionale K_p . Per questo motivo si preferisce far riferimento alla terna di coefficienti nella forma K_p , T_i , T_d (vedi Figura 10), piuttosto che alla terna K_p , K_i , K_d . Infatti, mettendo a fattor comune il principale coefficiente proporzionale, i restanti T_i e T_d possono essere più

stabilmente fissati da programma.

Nel caso si volessero controllare attuatori di potenza (per esempio riscaldatori o motori) è possibile collegare l'uscita PWM ad un buffer di potenza come per esempio il "Driver MOSFET PWM 15A 400W (codice 3085-MOSDRIVER). Infatti i pin ATtiny85 non possono fornire più di 50 mA.

HARDWARE PID

Come si vede dallo schema elettrico è presente anche una uscita per Servo-attuatore. Infatti, come vedremo più avanti, può pilotare dei Servo.

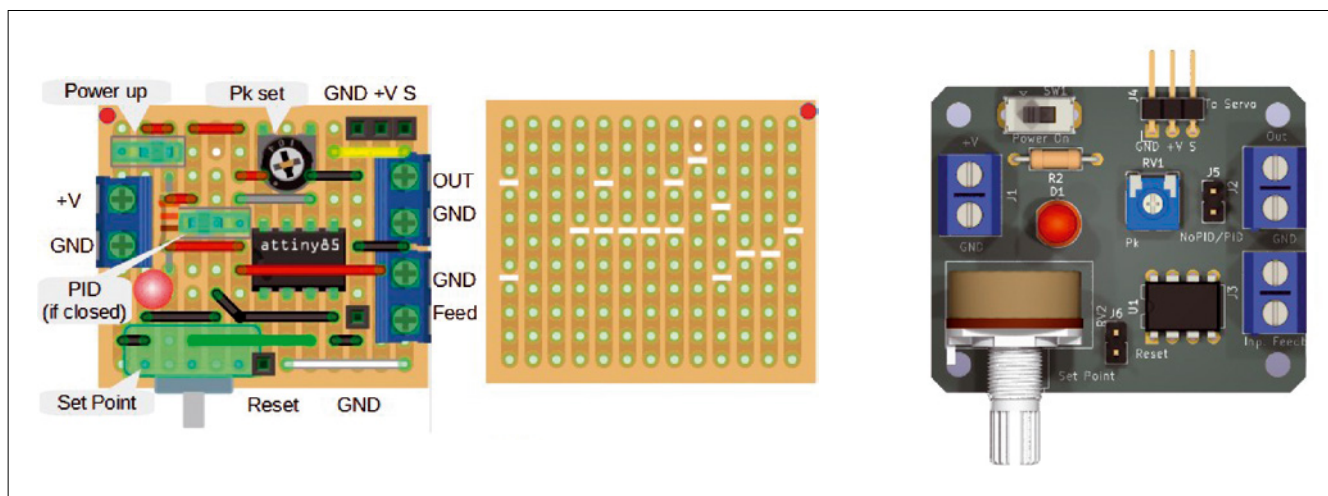
In **Fig. 11** sono invece visualizzati gli schemi fronte-retro realizzati con "stripboard". Anche in questo caso è fornita la soluzione PCB i cui file Gerber sono scaricabili dal sito della rivista.

SOFTWARE PID

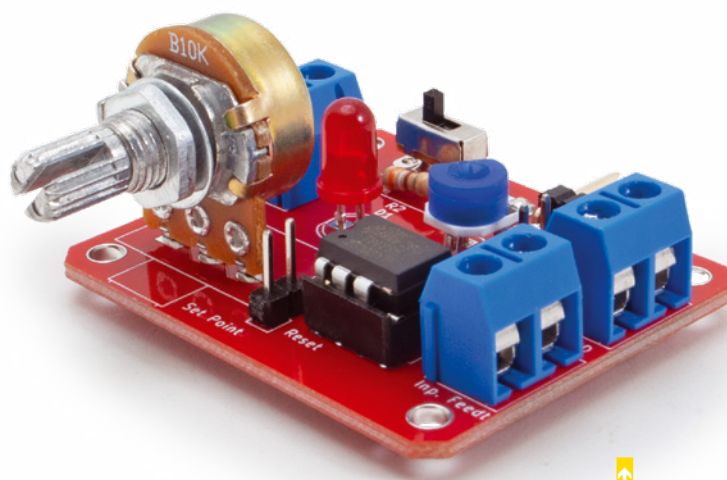
Principali aspetti del codice, nel caso di funzionamento PID:

- L'intervallo di "sampling" DT è fissato a 10 millisecondi (100 Hz), anche se la lettura analogica può arrivare a circa 30KHz, perché il PWM su powermos non può mantenere frequenze elevate. In caso contrario è possibile ridurlo a circa 0.5 millisecondi (2 KHz) che è approssimativamente il tempo di calcolo.
- Il Setpoint è normalizzato ai valori compresi tra 0 e 1
- K_p è regolabile con il trimmer (valore normalizzato tra 0 e 1). Quindi attualmente K_p è compreso tra 0 ed 1. Nel caso si voglia aumentarlo bisogna inserire un fattore di scala (ma bisogna tener presente che l'uscita è forzata nell'intervallo 0-1).

Fig. 11
Stripboard
fronte/retro
e rendering
PCB.



- T_i e T_d sono definiti da programma.
È comunque sempre possibile utilizzare, da programma, la forma K_p , K_i , K_d , per una inizializzazione alternativa.
- Anche la lettura del feedback è normalizzata tra 0 ed 1. La tensione letta deve essere tra 0 e V_{cc} . Per questo è previsto un fattore di scala che attualmente è pari all'unità.
Oppure bisogna provvedere con un adattamento hardware del livello del segnale di feedback.
- Ciclo: ogni DT secondi vengono effettuati i seguenti passi (funzione loop()):
 - legge i parametri: Setpoint, K_p e feedback
 - calcola l'output del PID (tra 0 e 1)
 - aggiorna l'output PWM fissandolo tra 0 e 255
- Se il PID è disabilitato, dopo aver letto il Setpoint lo converte semplicemente nel valore PWM (0-255).
L'output è considerato in fase con l'errore.
 - Ovvero se l'errore è positivo (Setpoint non ancora raggiunto), l'output deve essere positivo; ma se il calcolo PID sull'errore dà risultato negativo l'output non può che essere zero.
 In caso di processi inversi (per esempio: aumentare il raffrescamento se la temperatura è ancora alta e quindi l'errore è negativo) è sufficiente applicare un fattore di scala negativo al feedback.
In ogni caso l'output, sempre positivo, tende a zero quando l'errore tende a zero o è negativo.
- Per il calcolo PID sono stati confermati i seguenti suggerimenti:
 - La somma integrale è calcolata come somma ($k_i \cdot \text{err}$) e non come somma(err).



La scheda Driver Motore montata.

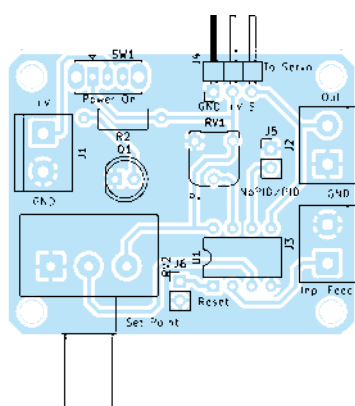
In questo modo le modifiche di K_p vengono inglobate ed attenuate nella sommatoria.

- La somma integrale è bloccata al valore massimo (1) o al valore minimo (0) in caso di superamento degli stessi (anti-windup).
- La derivata è calcolata come il negativo della derivata dell'input (feedback) e non dell'errore. Infatti se il setpoint non cambia i due valori coincidono.

In questo modo si riducono gli "spike" in caso di modifica del setpoint (anti-spike).

Anche in questo caso il sorgente del software per ATtiny85 si può scaricare dal sito della rivista con il nome **"TinyPID.ino"**.

piano di montaggio PID



Elenco Componenti:

- R1390 ohm
- RV1: Trimmer 5 mm 10 kohm
- RV2: Potenzimetro 10 kohm
- D1: LED 5 mm rosso
- SW1: Deviatore slitta miniatura

Varie:

- Morsetto 2 vie passo 5.08 mm (3 pz.)
- Strip maschio 2 vie passo 2.54 mm (2 pz.)
- Strip maschio 3 vie passo 2.54 mm 90°
- Circuito Stampato S1747 (47x38 mm)

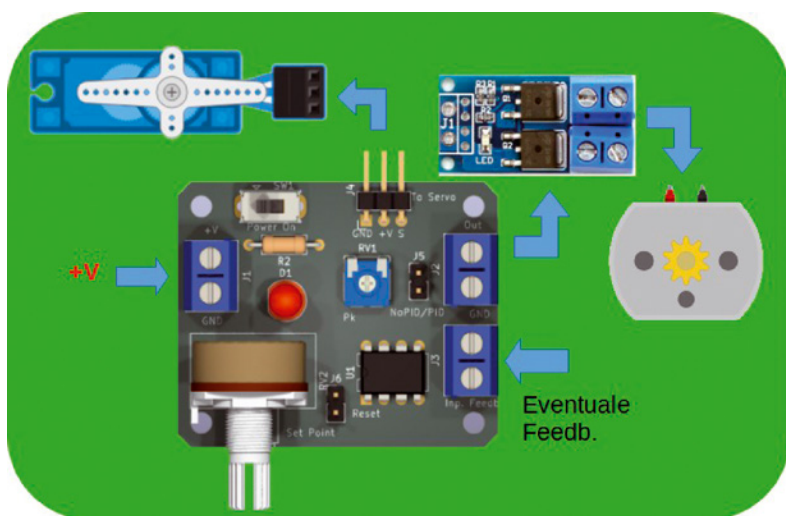


Fig. 12
Driver motore
o Driver Ser-
vo-attuatore.

PID PER CONTROLLO DI POSIZIONE

Il comportamento del PID descritto precedentemente è dedicato ad un azionamento di potenza che deve raggiungere e mantenere un certo risultato. Non può essere applicato come "inseguitore" di un target: per esempio per una funzione di "tracking" o una funzione di pilotaggio.

Per questo tipo di funzionalità occorre avere una uscita che possa variare in più o in meno rispetto ad un valore base che corrisponde all'errore zero (uscita bidirezionale). È possibile modificare l'output del PID precedente facendo corrispondere il valore 0.5 all'errore zero, ma in ogni caso occorrerebbe poi applicare un ponte H all'uscita.

Per una funzionalità di "tracking" basta modificare il PID per pilotare un Servo invece che un'uscita PWM. Per semplificare il software, si è preferito utilizzare una particolare modalità PWM piuttosto che caricare una libreria Servo.

Infatti, selezionando una frequenza di 125 Hz ovvero di 8 millisecondi di intervallo PWM, si

possono pilotare impulsi di lunghezza da 0.5 a 2.5 millisecondi (ogni 8 millisecondi) come previsto dalle specifiche servo. Il valore di uscita (0-1) sarà allora riparametrizzato tra i valori 15 e 80 PWM. Il PWM non sarà utilizzato nella sua completa escursione, ma la gestione risulta semplificata e sufficientemente precisa (2.7° per valore PWM).

MODIFICA DELLE FUNZIONALITÀ DEL PID

Come detto, il ponticello chiuso abilita il PID, altrimenti il modulo agisce come un semplice driver PWM, cioè come un controllo a catena aperta. Ma sono possibili anche altre modifiche di funzionamento senza riprogrammare ATtiny85 come nel caso dell'encoder. Poiché non sono a disposizione altri pin o pulsanti, per modificare il comportamento ci si è rivolti al pin di Reset. Quando viene chiuso a massa, il pin di Reset, oltre a produrre un restart, attiva un flag diverso da quello per un restart da interruzione di alimentazione. Quindi è possibile verificare questa condizione ed utilizzarla per riconfigurare il software salvando la modalità in EEPROM per i futuri restart. In particolare quando si resetta (Reset collegato a 0):

- dalla modalità di default (sia PWM che PID) si passa alla modalità bidirezionale (ovvero output=0.5 per errore zero);
- poi alla modalità Servo-Attuatore;
- da quest'ultima si torna alla modalità di default.

CONCLUSIONI

L'articolo ha illustrato come il microcontrollore ATtiny85, con le sue dimensioni ridotte e costi contenuti, rappresenti una soluzione versatile per diverse applicazioni elettroniche. Abbiamo esplorato in dettaglio l'integrazione di un encoder rotativo e di un controllore PID, evidenziando i vantaggi dell'utilizzo del collegamento I2C per gestire in modo efficiente le risorse del sistema.

Grazie all'adozione di tecniche avanzate come l'uso degli interrupt per la gestione dell'encoder e le ottimizzazioni nel calcolo PID, è possibile migliorare significativamente le prestazioni dei dispositivi elettronici, riducendo al contempo la complessità del software necessario. Inoltre, la possibilità di configurare e aggiornare il microcontrollore rende l'ATtiny85 una scelta pratica e flessibile per sviluppatori e hobbisti.

In sintesi, l'ATtiny85 si conferma come un componente chiave per progetti di automazione e controllo, offrendo un equilibrio ideale tra potenza di calcolo, semplicità d'uso e possibilità di personalizzazione.



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
ATTINY85	Microcontrollore Atmel ATTINY85-20PU	4,90
ENCODER12MM	Encoder rotativo 12 mm con manopola	3,50
MOSDRIVER	Driver MOSFET PWM 15A 400W	2,90

IL MATERIALE VA RICHIESTO A:

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

COMUNICAZIONE **WIRELESS** FLESSIBILE

di **GAMAL LABIB**

L'utilizzo di una memoria EEPROM consente al modulo ESP di connettersi facilmente a reti prememorizzate.

Quando si connette un ESP8266 a una rete Wi-Fi, potrebbe essere utile adottare un approccio più flessibile rispetto all'uso di credenziali WLAN hard-coded e fisse. Offriamo una soluzione con un insieme di reti AP preferite da scegliere in modo interattivo. Inoltre, possiamo sfruttare la funzionalità Wi-Fi Protected Setup (WPS) supportata da molti AP e router.

L'ESP8266 è microcontrollore (MCU) molto popolare che supporta la comunicazione wireless e trova applicazione in vari ambiti dell'Internet of Things (IoT). Le schede basate su questo modulo solitamente hanno credenziali dell'Access Point (AP), come il Service Set Identifier (SSID) e la passphrase, codificate nei loro sketch Arduino. Tuttavia, queste impostazioni possono causare problemi di connessione quando la topologia della rete locale wireless (WLAN) cambia. Diventa un incubo dover tornare all'IDE di Arduino solo per aggiornare gli sketch sulle schede MCU già distribuite. In questo articolo, dimostrerò una soluzione che offre flessibilità nell'unire una scheda MCU a una WLAN. Invece di codificare un unico set di credenziali WLAN negli sketch del progetto, perché non avere un insieme di reti AP preferite da selezionare interattivamente? Stabilendo un dialogo interattivo tra l'utente e la scheda MCU attraverso un touchscreen, una pagina web o mezzi simili, diventa possibile specificare nuove impostazioni WLAN per la scheda MCU in

tempo reale. Inoltre, possiamo sfruttare la funzione Wi-Fi Protected Setup (WPS) supportata da molti AP e router. Tuttavia, sorge una domanda: dobbiamo eseguire i passaggi per connettersi tramite dialogo interattivo o WPS ogni volta che la scheda MCU si riavvia? La risposta è no, purché manteniamo le nuove impostazioni WLAN specificate in una memoria non volatile accessibile alla MCU. Fortunatamente, l'ESP8266 dispone di una memoria interna EEPROM che può essere manipolata dal codice utente per memorizzare e recuperare dati. Ho utilizzato questa funzionalità per memorizzare fino a 10 set di credenziali WLAN impostati con uno dei metodi sopra menzionati. Questo approccio non solo offre flessibilità di rete, ma consente anche alla scheda MCU di connettersi alla WLAN con la migliore copertura tra le 10 memorizzate, se lo sviluppatore decide di farlo. Tuttavia, è importante notare che scrivere frequentemente sulla EEPROM interna non è consigliabile, poiché la vita di una EEPROM dipende dal numero di cicli di scrittura che subisce. Per aggirare questa limitazione, mostrerò come collegare una EEPROM esterna, pur utilizzando quella interna per i vari esempi.

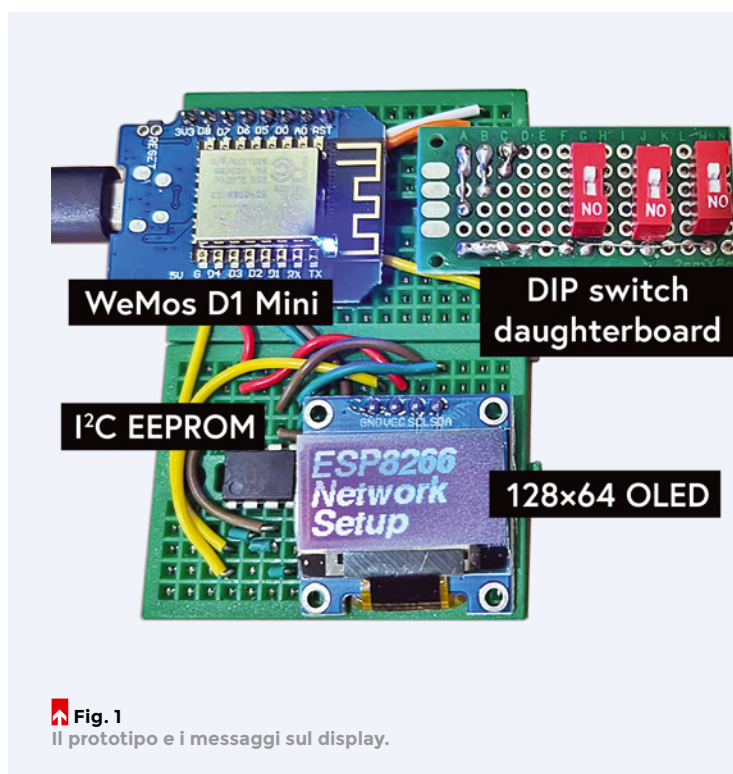


Fig. 1
Il prototipo e i messaggi sul display.



HARDWARE

La Bill of Materials (BOM) per questo progetto è piuttosto compatta. Ho utilizzato il WeMos D1 Mini basato su ESP8266, noto per la sua facilità d'uso con l'IDE di Arduino e per il suo ingombro ridotto. Un modulo display grafico OLED da 0,9" con una risoluzione di 128x64 pixel per visualizzare messaggi informativi e di debug connesso al bus I2C (pin D1/SCL e D2/SDA) insieme a un chip EEPROM esterno da 8 KB.

Per selezionare una delle otto modalità operative della MCU (vedi **Tabella 1**), ho utilizzato una combinazione di tre DIP switch, collegati a tre dei GPIO digitali del WeMos (D5, D6 e D7). Poiché i connettori dei DIP switch sono molto piccoli e non si adattano alla breadboard, ho costruito una scheda daughter board per ospitarli. La realizzazione del progetto su breadboard e gli screenshot del WeMos in azione sono mostrati in **Fig. 1**.

SOFTWARE

Il progetto pone l'accento sulle migliori strategie per lo sviluppo software, come la scrittura di codice ben strutturato e riutilizzabile. Per raggiungere questo obiettivo, ho suddiviso lo sketch di 800 linee in sei file separati, ciascuno dei quali gestisce un componente hardware o una funzionalità specifica (vedi **Tabella 2**).

Questo approccio rende il codice più gestibile e aiuta i lettori a comprenderlo. Il file principale dello sketch si concentra sulle modalità di rete del WeMos all'interno della funzione setup(), lasciando la funzione loop() libera per il codice applicativo del WeMos.

Durante la fase di sviluppo, ho utilizzato loop() per manipolare il lampeggio del LED integrato e assicurarmi che il WeMos funzionasse correttamente.

Mode	DIP 1	DIP 2	DIP 3	WeMos Action
1	0	0	0	Default WLAN selected
2	0	0	1	Scan available WLANs
3	0	1	0	Clear internal EEPROM
4	0	1	1	Dump internal EEPROM
5	1	0	0	Initialize internal EEPROM
6	1	0	1	Check external EEPROM
7	1	1	0	WPS
8	1	1	1	Interactive WLAN setup

Tabella 1
Configurazione dei DIP switch per la scelta della modalità di funzionamento.

UTILIZZO DELLA EEPROM PER IL NETWORKING WLAN

Il modulo ESP8266 dispone di 4 MB di memoria flash, di cui 4 KB sono riservati per emulare una EEPROM normalmente integrata nei dispositivi Arduino. Nel nostro esempio, lo sketch memorizza di default le credenziali AP funzionanti più recenti nella EEPROM interna. Utilizzando una memoria non volatile interna (o esterna), il nostro sketch può richiamare le credenziali AP ogni volta che si avvia dopo un riavvio. La EEPROM è accessibile tramite chiamate di funzione dalla libreria EEPROM di Arduino, e seguirò questa direzione per salvare i dati di rete che devono essere non volatili quando il modulo è spento. Il concetto principale è mantenere la struttura proposta mostrata nella **Tabella 3** per contenere le credenziali WLAN AP, fino a quante la EEPROM interna o esterna può contenere. Ogni voce successiva nella struttura contiene l'SSID dell'AP o la sua passphrase in 32 e 64 byte di memoria, rispettivamente. Sono quindi 96 byte in totale per AP. Ogni volta che l'utente aggiunge una nuova WLAN all'elenco delle reti preferite, il codice aggiunge a questa struttura le credenziali dell'SSID e della passphrase della nuova rete. L'utente deve cancellare la EEPROM alla prima esecuzione del codice del progetto, utilizzando la Modalità 3, segnando così la fine dell'elenco con voci vuote. L'utente avrà la possibilità di applicare le credenziali WLAN codificate, utilizzando la Modalità 1, oppure di controllare la connettività utilizzando le credenziali nella EEPROM, operazione effettuata utilizzando la Modalità 8. La mia implementazione per questo progetto si limita a questo meccanismo per un massimo di 10 WLAN, limitato dalla costante definita MEMCNT nel codice, oltre a cancellare la struttura e iniziarla con un elenco preimpostato di credenziali AP a scelta dell'utente. Il lettore può aumentare o diminuire la dimensione di MEMCNT secondo necessità. Le conoscenze di base per accedere alla EEPROM possono essere trovate in www.aranacorp.com/en/using-the-eeprom-with-the-esp8266. Ora agisco sulle modalità operative del WeMos, ognuna delle quali richiede la configurazione dei DIP switch (come da **Tabella 1**) e poi riavvio del WeMos per applicare la configurazione.

Modalità 1: Impostazioni WLAN Predefinite (Hard-Coded)

Una MCU richiede credenziali WLAN, composte da un SSID e una passphrase per il relativo access

.ino file	Description	Line Count
wemos-d1-mini-network	Global variables, setting up network mode	240
wlan-utils	Manipulate WLAN connections: WPS setup, scan/list WLANs, check connectivity	110
tft-utils	Manipulate TFT display	130
internal-eeprom-utils	Initialize EEPROM with five preset WLAN credentials, reset EEPROM, read/write entries	180
external-eeprom-utils	Check external EEPROM connectivity and size	40
ap-web-setup	Interactive WLAN setup: Initiate web server, display detected WLAN SSIDs, save selected WLAN to EEPROM	100

 **Tabella 2** Suddivisione dei file del codice Arduino e delle loro funzioni.

Entry Size (Bytes)	Description	Type
2	AP entry count	integer
32	AP1 SSID	string
64	AP1 Passphrase	string
.	.	.
.	.	.
.	.	.
32	APn SSID	string
64	APn passphrase	string

 **Tabella 3** Gestione della EEPROM per le impostazioni di più AP.

point. Le credenziali WLAN possono essere codificate direttamente nello sketch caricato o passate interattivamente alla MCU quando l'utente è dotato di una tastiera e di un display o di un Serial Monitor nell'IDE collegato tramite la porta seriale. In questo progetto, utilizzo il metodo convenzionale di fissare le credenziali dell'AP direttamente nello sketch.

Modalità 2: Scansione WLAN

In questa modalità, il WeMos si disconnette da qualsiasi WLAN a cui è connesso per poter scansionare altre reti accessibili. Le WLAN rilevate vengono elencate con i loro SSID e le intensità del segnale in decibel (dB). Questa modalità fornisce al lettore una panoramica sulle possibilità di connettere la scheda WeMos alla rete e quale modalità

```
14:34:00.668 -> scan available wlangs
14:34:05.788 -> Disconnecting previously connected WiFi
14:34:08.148 -> scan completed
14:34:08.148 -> 1 Networks found
14:34:08.148 -> 1: GGGG (6) {-63}
14:34:08.268 ->
```

Fig. 2
Serial Monitor dell'IDE di Arduino – Scansione della rete wireless in Modalità 2.

scegliere per unirsi a una WLAN.

La **Fig. 2** mostra il risultato dell'avvio in questa modalità, dove è stata rilevata solo una WLAN.

Modalità 3: Cancellazione della EEPROM Interna

Questa modalità fa sì che il WeMos cancelli la sua EEPROM interna, preparando il dispositivo per la creazione di un nuovo elenco di WLAN preferite. Tutti i byte nella EEPROM vengono azzerati a 0x00 in questa modalità.

Modalità 4: Visualizzazione della EEPROM Interna

Questa modalità fa sì che il WeMos mostri la struttura delle credenziali WLAN, come spiegato nella **Tabella 3**, dalla EEPROM interna. In questa modalità, viene stampata una serie di SSID e passphrase formattati per le credenziali degli AP memorizzati.

Modalità 5: Inizializzazione della EEPROM Interna

Questa modalità consente all'utente di preimpostare le credenziali di 10 WLAN preferite nella EEPROM interna. Questa azione rappresenta un'estensione rispetto alla tradizionale codifica di credenziali di una singola WLAN. Al riavvio in Modalità 8, il WeMos esaminerà l'elenco delle credenziali per le connessioni WLAN. Se i tentativi di connessione a una delle WLAN preferite falliscono, l'utente dovrà ricorrere alle modalità interattive o WPS.

Fig. 3
Serial Monitor – Verifica dell'integrità della EEPROM esterna in Modalità 6.

```
14:35:43.671 -> check external eeprom
14:35:48.751 -> Disconnecting previously connected WiFi
14:35:48.871 -> check external eeprom
14:35:48.871 -> External eeprom isConnected with Status:
14:35:48.871 ->
14:35:48.871 -> TEST: determine size of external eeprom
14:35:48.871 -> 80      FF
14:35:48.871 -> 100     0
14:35:48.871 -> 200     0
14:35:48.871 -> 400     0
14:35:48.871 -> 800     FF
14:35:48.911 -> 1000    AA
14:35:48.911 -> 2000    AA
14:35:48.911 -> external eeprom size: 8192 Bytes
```

Modalità 6: Verifica dello Stato della EEPROM Esterna

In questa modalità, il WeMos verifica l'integrità della EEPROM esterna, nel mio caso una 24LC64 con 8 KB di memoria. La **Fig. 3** mostra l'output di questo controllo sul Serial Monitor. Non ho coinvolto ulteriormente questo dispositivo esterno nel progetto poiché l'utilizzo della EEPROM interna è minimo. Il lettore può fare riferimento al codice I2C EEPROM playground.arduino.cc/Code/I2CEEPROM e alla libreria github.com/RobTillaart/I2C_EEPROM per l'integrazione nel progetto.

Modalità 7: Connessione Diretta alla WLAN (WPS)

Il WPS è una funzionalità supportata dai moderni Access Point (AP) che semplifica il processo di connessione dei dispositivi wireless a una WLAN. Piuttosto che fornire all'MCU le credenziali dell'AP, come SSID e passphrase/pre-shared key, l'MCU può fornire il pin di sicurezza dell'AP o il proprio pin preimpostato (vedi **Fig. 4**). In alternativa, è possibile premere un pulsante WPS sull'AP per la connessione diretta all'MCU richiedente (vedi **Fig. 5**). La **Fig. 6** mostra il risultato della connessione stabilita utilizzando il WPS.

In questo progetto, limito la connessione WPS all'alternativa del pulsante AP poiché rende il codice più semplice e generico, adattandosi a qualsiasi AP raggiungibile, come segue:

```
bool wpsSuccess = WiFi.beginWPSConfig();
if (wpsSuccess) {
    String newSSID = WiFi.SSID();
    if (newSSID.length()==0) { wpsSuccess = false; }
}
```

La scheda WeMos deve essere nelle vicinanze dell'AP scelto per registrarne i parametri nella EEPROM. Riconnettere il modulo allo stesso AP non richiederà più la pressione del pulsante WPS, poiché abilitiamo l'ESP a recuperare i parametri dell'AP memorizzati per il riutilizzo, come segue:

```
WiFi.mode(WIFI_STA);
WiFi.begin(WiFi.SSID().c_str(),WiFi.psk().c_str());
```

Si noti che il modulo ESP8266, che costituisce il cuore della scheda WeMos, memorizza automaticamente le credenziali di rete recenti con successo nella EEPROM interna.

Pertanto, dobbiamo posizionare la nostra struttura di gestione, descritta nella **Tabella 3**, in un'altra

☒ Enable WPS

WPS configuration

☒ WPS by push button (PBC)
 ☐ WPS by client PIN (PIN)
 ☐ WPS by AP-PIN (AP-PIN)

WPS configuration

☐ WPS by push button (PBC)
 ☒ WPS by client PIN (PIN)
 ☐ WPS by AP-PIN (AP-PIN)

PIN:

WPS configuration

☐ WPS by push button (PBC)
 ☐ WPS by client PIN (PIN)
 ☒ WPS by AP-PIN (AP-PIN)

AP-PIN:

Fig. 4 Configurazioni AP per le alternative WPS.

parte della EEPROM per evitare conflitti con gli altri meccanismi della MCU.

Modalità 8: Configurazione Interattiva della WLAN

In questa modalità, il WeMos verifica innanzitutto l'integrità delle credenziali memorizzate nella EEPROM. Se la MCU riesce a connettersi a un AP, la modalità termina e la funzione loop() prende il controllo. In caso contrario, la MCU entra in modalità interattiva, in cui scansiona le WLAN nelle vicinanze e avvia un web server con l'indirizzo IP 192.168.4.1 per visualizzare gli SSID degli AP rilevati. L'utente deve configurare l'adattatore WLAN del proprio computer con impostazioni IP fisse per la rete privata del WeMos, poiché il WeMos non

fornirà alcun servizio DHCP. Una pagina web simile a quella mostrata nella **Fig. 7** mostrerà i risultati della scansione e consentirà all'utente di scegliere la rete preferita. L'utente quindi clicca sul pulsante "Submit Query" per salvare le credenziali WLAN nella EEPROM e stabilire una connessione con la WLAN selezionata.

Abbiamo visto come le EEPROM possono fare la differenza quando si tratta di collegare una MCU wireless. Invece di codificare le credenziali WLAN nel codice, l'utilizzo di una EEPROM permette di mantenere e aggiornare dinamicamente le credenziali di più AP. Il codice sorgente è modulare e facile da comprendere e può essere riutilizzato per la gestione delle EEPROM, delle WLAN e per la comunicazione con la MCU tramite il web.

Per maggiori informazioni visita la pagina elektormagazine.com/magazine/elektor-312/62149



Fig. 5 Pulsante WPS e indicatore di un tipico AP.

```

14:37:03.552 -> Disconnecting previously connected WiFi
14:37:03.632 -> Try wps if necessary
14:37:03.632 ->
14:37:03.632 -> Could not connect to WiFi ... state= '7'
14:37:03.672 -> Please press WPS button on your router
14:37:03.672 -> WPS config start
    
```

Fig. 6 Serial Monitor – Configurazione WPS in Modalità 7.

File Edit View History Bookmarks Tools Help

192.168.4.1/ × +

← → ↻

192.168.4.1

ESP8266 WiFi Connectivity Setup 192.168.4.1

1. GGGG (-57)

SSID:

Fig. 7 Modalità 8, la selezione interattiva della WLAN.



ENTRA NEL MONDO DELL'ELETTRONICA

KIT FOR MAKERS

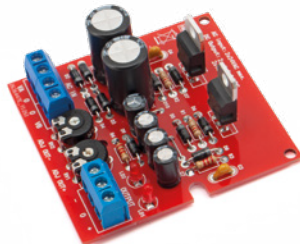
Le scatole di montaggio di Futura Elettronica sono semplici da montare e da utilizzare, ideali per avvicinarsi al fantastico mondo dell'elettronica. Non utilizzano microcontrollori e sono fornite con schemi elettrici e istruzioni.

Alimentatore stabilizzato da 1,5 a 35V



cod. FK003 € 10,50

Alimentatore stabilizzato duale e regolabile



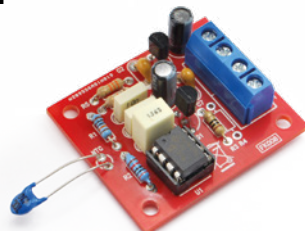
cod. FK004 € 14,00

Scheda di controllo automatico per ventilazione



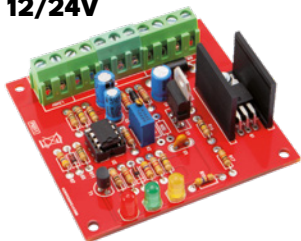
cod. FK007 € 14,00

Regolatore di velocità per ventole



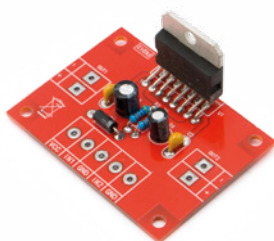
cod. FK008 € 17,80

Regolatore PWM per pannelli fotovoltaici 12/24V



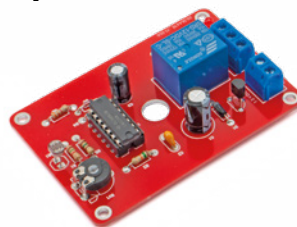
cod. FK011 € 18,00

Amplificatore audio stereo 2x15



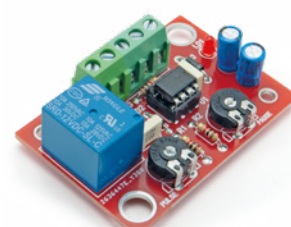
cod. FK012 € 8,50

Interruttore crepuscolare



cod. FK001 € 9,00

Modulo timer a intermittenza



cod. FK002 € 12,00

Timer Start e Stop



cod. FK005 € 14,00

Interruttore battimani



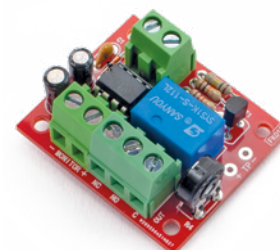
cod. FK006 € 14,00

Luce psichedelica a LED



cod. FK009 € 12,00

Protezione bassa tensione



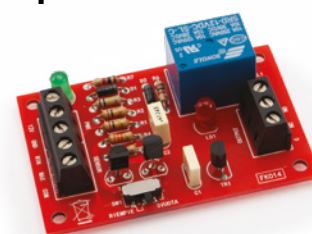
cod. FK010 € 11,50

Automatismo per tende e tapparelle motorizzate



cod. FK013 € 13,50

Stabilizzatore livello liquidi



cod. FK014 € 13,50

Prezzi IVA inclusa

CON I KIT DI FUTURA ELETTRONICA

Indicatore di livello liquido per vasche



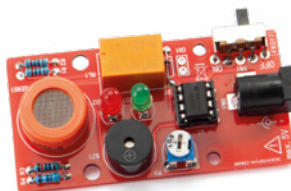
cod. FK015 € 8,50

Allarme batteria scarica



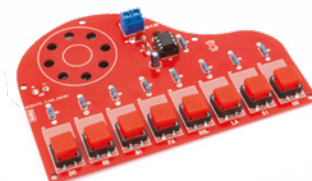
cod. FK016 € 8,00

Alcol tester



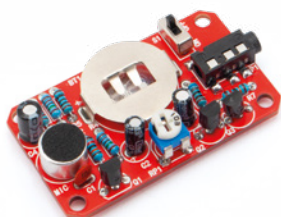
cod. FK017 € 9,00

Mini piano elettronico con NE555



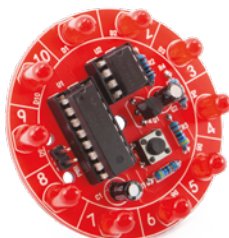
cod. FK018 € 8,00

Amplificatore acustico



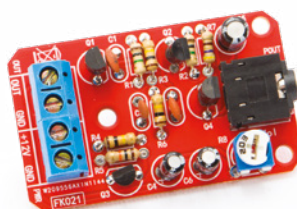
cod. FK019 € 9,00

Ruota della fortuna a LED



cod. FK020 € 8,00

Generatore di rumore bianco



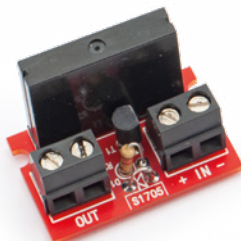
cod. FK021 € 8,00

Candela a LED con effetto fiamma



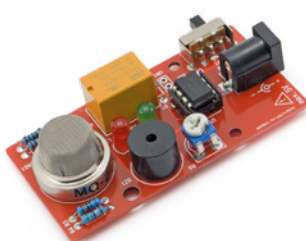
cod. FK022 € 8,00

Scheda con relè allo stato solido



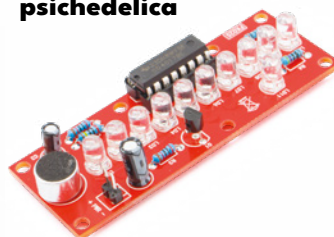
cod. FK023 € 5,50

Scheda con sensore di gas GPL, propano e idrogeno



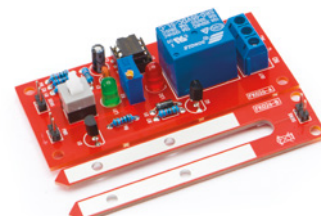
cod. FK024 € 11,00

Barra a LED psichedelica



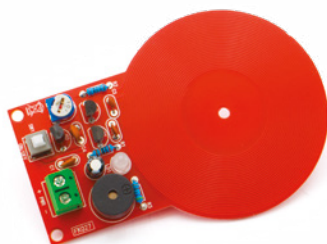
cod. FK025 € 8,50

Sensore di umidità



cod. FK026 € 13,00

Mini Metal Detector



cod. FK027 € 7,50

Lampeggiatore a LED



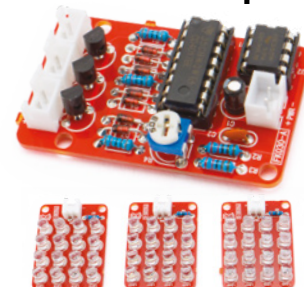
cod. FK028 € 6,50

Campanello



cod. FK029 € 8,00

Luci LED stroboscopiche



cod. FK030 € 14,00

Saldare diventa più facile!

Supporto per saldatura con 6 bracci flessibili e accessori



cod. **HR2120**
€ 39,00

Composta da lampada a 16 LED con lente d'ingrandimento 3X, terza mano, braccio flessibile da 25 cm con clip a coccodrillo, porta saldatore e porta rocchetto di stagno. Viene fornita completa di pasta per la saldatura (10g) e di spugnetta per rimuovere lo stagno dalla punta del saldatore.

Supporto per saldatura composto da 6 bracci flessibili con clip a coccodrillo, lente d'ingrandimento 3x, ventola da 30X30 mm per assorbire il fumo della saldatura con alimentazione USB.

Postazione multifunzione



cod. **TIP019**
€ 48,00

Supporto per saldatura a 4 bracci flessibili con clip a coccodrillo



cod. **SHH4ARMS**
€ 23,90

Ideale per saldare e dissaldare componenti elettronici o per eseguire piccoli lavori di precisione. Con spugnetta e un pulisci punta per saldatore con paglietta in ottone.

Terza mano con lente, luce e portasaldatore



cod. **TP232**
€ 17,00

Terza mano con luce LED, lente d'ingrandimento 3,5X e supporto per saldatore

Terza mano con lente d'ingrandimento



cod. **VTHH**
€ 7,90

Terza mano con lente d'ingrandimento 2,5X e supporto per saldatore

Supporto per serraggio scheda

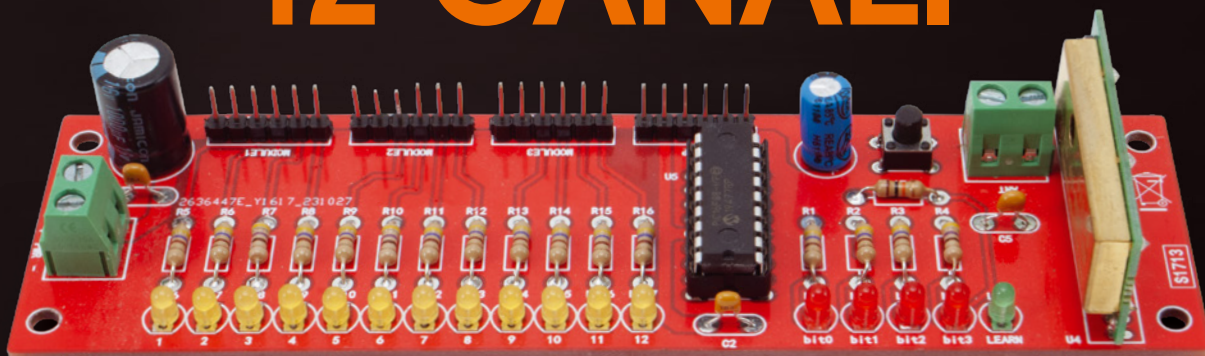
Permette di sorreggere e bloccare, tramite due morsetti che possono ruotare di 360°, una scheda elettronica con dimensioni max 20X14 cm.



cod. **VTHH6**
€ 12,00

Prezzi IVA inclusa

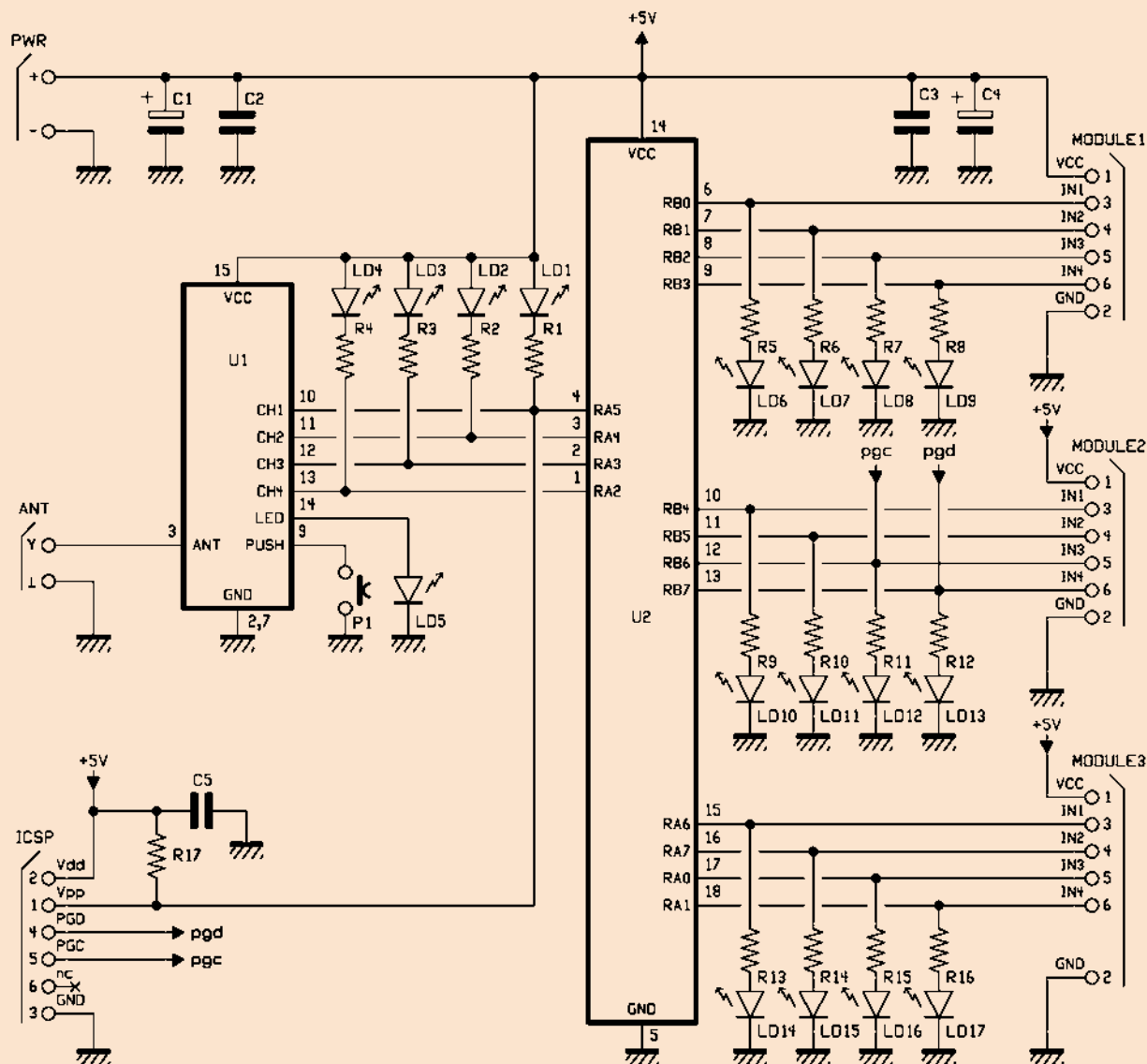
RADIOCOMANDO 12 CANALI



di **CRISTIANO RUGGERI**

Dotato di 12 uscite a relè azionabili da telecomandi a 4 e a 12 tasti, questo ricevitore può funzionare sia in modo tradizionale a singolo relè, che in modo "fermodellismo" a combinazione di sequenze.

Normalmente, qualsiasi nostro progetto nasce da un'idea ben precisa di applicazione, applicazione che tendenzialmente vuole essere la più generalizzata possibile, affinché chiunque possa fruirne nel modo che più gli è congeniale. Questo progetto di ricevitore, invece, ha avuto una gestazione alquanto bizzarra, poiché è stato esplicitamente pensato per un'applicazione molto di nicchia, ma poi è stato dirottato/adattato anche verso altri impieghi: l'idea di base, infatti, era poter controllare molto facilmente ed in modo automatico un diorama ferroviario dotato di 4 semafori, 4 binari di stazionamento e 2 coppie di scambi, attivando automaticamente 12 relè secondo sequenze prestabilite con le quali gestire in modo opportuno ogni semaforo, il relativo binario di stazionamento e lo



scambio quando necessario. Tutto ciò con lo scopo di evitare all'operatore eventuali errori di comando (inevitabili quando si attua una gestione completamente manuale) che talvolta non causano effetti e talvolta invece provocano deragliamenti spesso distruttivi, poiché anche un trenino giocattolo che deraglia su un plastico finisce col travolgere e anche distruggere ciò che trova sulla sua traiettoria. Le 12 uscite a relè, inoltre, dovevano essere azionate tramite un telecomando a 12 canali modello TX-12CH della Aurel, in cui ogni tasto avrebbe avviato una sequenza di comandi specifica. Purtroppo il trasmettitore (**Fig. 1**), oltre ad avere un costo piuttosto importante, è divenuto nel frattempo di difficile reperibilità, quindi questa nuova necessità ci ha portati all'aggiunta della

possibilità di gestire il ricevitore anche per mezzo di un telecomando a 4 pulsanti modello 8110-TX-4M-HCS (**Fig. 2**) del quale viene analizzato non solamente il tasto premuto, bensì anche la durata della pressione, così da poter discriminare tra durate brevi, medie e lunghe, e gestire le 12 uscite a relè a 3 gruppi di 4, in base alla durata del comando stesso. Giunti al prototipo funzionante, che soddisfaceva appieno la specifica di progetto, abbiamo constatato quanto fosse (come già anticipato) un prodotto troppo specializzato e di nicchia: non sappiamo, infatti, quanti dei nostri lettori si dilettono di elettronica e contemporaneamente di fermodellismo vintage, magari di tipo H0 analogico vecchia scuola piuttosto che "12 V realizzato a mattoncini Lego®".

Ecco quindi che abbiamo aggiunto la possibilità di far funzionare la ricevente in modo tradizionale, ossia facendo corrispondere l'attuazione di un relè specifico a fronte di un tasto specifico premuto sul telecomando.

La versione finale del progetto, che presentiamo in queste pagine oggi, è pertanto in grado di operare con telecomandi sia a 4 che a 12 tasti, sia in modo tradizionale che in modo "fermodellismo". Per non farci mancare nulla, infine, abbiamo anche aggiunto la possibilità di cambiare il modo di funzionamento premendo un pulsante del telecomando durante la fase di accensione.

Se in qualche modo vi abbiamo incuriosito, o se siete alla ricerca di un telecomando a 12 canali per i vostri treni piuttosto che per le vostre apparecchiature controllabili da remoto, proseguite nella lettura di questo progetto che potremmo definire più diagonale che trasversale.

SCHEMA ELETTRICO

Come già anticipato, per semplificare il sistema e renderlo più compatto ed economico possibile, si è scelto di utilizzare tre schede a quattro relè, modello 2846-RELAY4CH a 5 V (possono essere facilmente reperite presso il sito www.futurashop.it) quindi alla fine tutto il sistema si riduce ad una ricevente HCS301, un microprocessore PIC16F88 e pochi altri componenti passivi. Vista la grande disponibilità di adattatori di rete stabilizzati a 5Vdc, non abbiamo previsto nemmeno un circuito regolatore di tensione a 5 V.

Iniziamo la descrizione partendo dalla decodifica radio, modello 8110-RX-4MHCS-4B di casa Aurel, indicata nello schema come U1: questo componente è già stato utilizzato altre volte per altri progetti ma, poiché "repetita juvant", spendiamo comunque qualche parola a riguardo. Si tratta di un ricevitore supereterodina operante in Ampiezza Modulata (AM) alla frequenza di 433,92 MHz, con a bordo la decodifica HCS301 con codifica Keeloq a codice variabile.

Questo ricevitore dispone di uscite a decodifica binaria: in pratica sulle sue quattro uscite (CH1+CH4, pin 10+13) viene presentato un codice binario che varia da "1111" quando nessun telecomando è stato attivato, fino a "0000" qualora fosse abbinato ad esso un ipotetico telecomando a 15 pulsanti, uno dei quali pigiato. Poiché le linee di uscita sono open collector al negativo, esse necessiterebbero di quattro resistenze di pull-up che forniscano il livello logico 1 quando le uscite della decodifica sono a riposo: questa è la ragione per cui la logica

di uscita prevede 1111 a riposo e 0000 con tutte le uscite attive. Dal punto di vista software, la cosa non determinerà grossi problemi in quanto per ottenere il conteggio normale a salire è sufficiente sottrarre il valore della lettura delle uscite alla costante 15, per cui a riposo avremo "15 - 15 (1111 in binario) = 0 (0000 in binario)".

U1 è stato inoltre completato con un pulsante di avvio della procedura di accoppiamento di telecomando, collegato alla linea PUSH (pin 9), un led verde di segnalazione di attività (LD5, uscita LED) e quattro led rossi, ognuno colla propria resistenza di limitazione da 470 ohm, collegati alle uscite CH1+CH4 in configurazione pull-up, così da fornire sia il livello logico alto a riposo che un'indicazione visiva dello stato delle uscite decodificate.

Le quattro uscite di U1 arrivano alle linee RA5, RA4, RA3, RA2 (pin 4, 3, 2, 1) del microcontrollore modello PIC16F88 (U2) che provvede a leggerle, decodificarle ed elaborare l'output verso le uscite di comando relè.

La scelta di questo microcontrollore è determinata dal fatto che dispone di tutto ciò che occorre senza sprechi ossia: in 18 pin DIL offre un oscillatore interno da 8 MHz (velocità anche eccessiva per il tipo di operazioni richieste, ma c'è! Quindi perché non sfruttarla?) dimensioni di memoria RAM e FLASH sufficienti per ospitare il codice sviluppato da noi e a lasciare spazio ad eventuali vostre aggiunte, 16 linee di I/O esattamente quante ne occorrono a noi. Inoltre è facilmente reperibile presso Futura Elettronica ad un costo ragionevole ed è di relativamente facile impiego, se si ha un minimo di dimestichezza con il PicBasic di casa Melabs.

Comunque delle 16 linee di I/O disponibili, quattro le abbiamo usate per leggere la decodifica e le rimanenti 12 sono state raggruppate per poter comandare fino a tre moduli a 4 relè. Ogni uscita, infine, è corredata di un led e relativa resistenza di limitazione, ed è comandata dal positivo, così da avere l'indicazione di relè attivo a bordo.

Completano il circuito tre connettori per il collegamento delle schede a relè (MODULE1, MODULE2, MODULE3), connettori questi che rispettano esattamente la sequenza dei pin delle schede a relè, il connettore per la programmazione in-circuit del microcontrollore (ICSP), la morsettiera per il collegamento dell'antenna (ANT), il morsetto di alimentazione a 5 V (PWR) con i condensatori da C1 a C4, cui è affidato il compito di filtrare eventuali disturbi provenienti dall'alimentatore esterno.



Elenco Componenti:

R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11,

R12, R13, R14, R15, R16: 470 ohm

R17: 10 kohm

C2, C3, C5: 100 nF ceramico

C4: 1000 µF 16 VL elettrolitico

C1: 220 µF 16 VL elettrolitico

LD5: LED 3 mm verde

LD1, LD2, LD3, LD4: LED 3 mm rosso

LD6, LD7, LD8, LD9, LD10, LD11, LD12, LD13, LD14,

LD15, LD16, LD17: LED 3 mm giallo

P1: Microswitch

U1: RX-4MHCS-4B

U2: PIC16F88-I/P (MF1713)

ANT, PWR: Morsetto 2 vie passo 5.08 mm (2 pz.)

ICSP, MODULE1: Strip maschio 6 vie

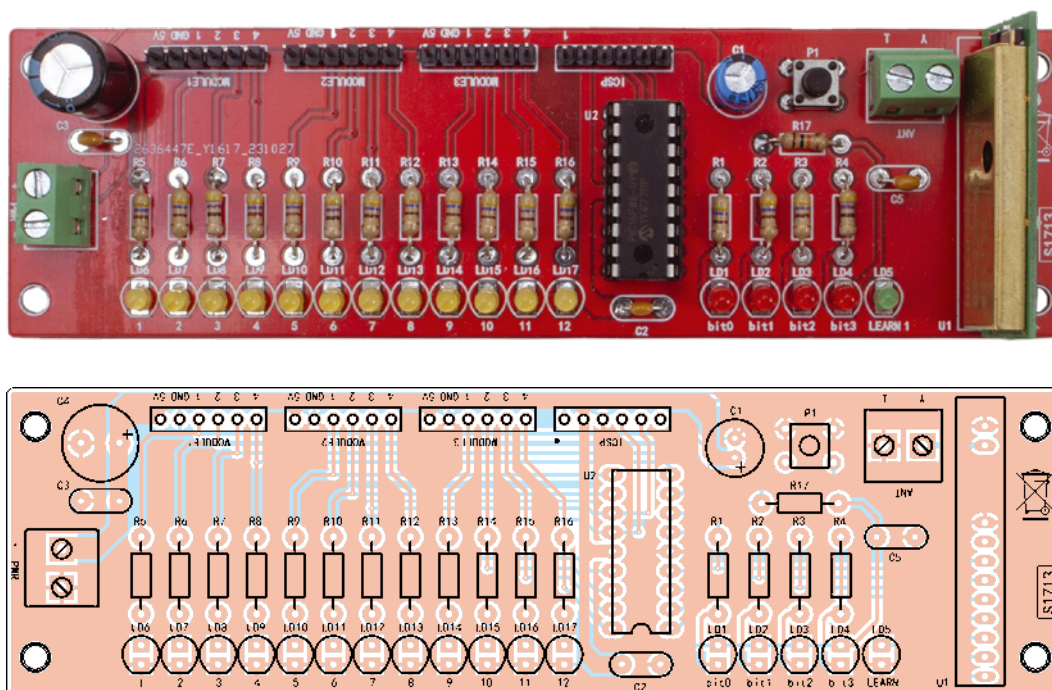
MODULE2: Strip maschio 6 vie

MODULE3: Strip maschio 6 vie

Varie:

- Zoccolo 9+9

- Circuito stampato S1713 (140x55mm)



REALIZZAZIONE PRATICA

Come sempre, per prima cosa occorre disporre del circuito stampato: a tale proposito potete scaricare il master a doppia faccia dal sito della Rivista, quindi realizzarlo con la fotoincisione, col press-n-peel, disegnandolo a mano, oppure (perché no?) usufruire del servizio di prototipazione circuiti stampati offerto da Futura Elettronica PCBPRODUCTION. Una volta ottenuto lo stampato, potete iniziare il montaggio dai componenti più bassi come resistenze, led e condensatori di piccola dimensione. Saldare ora lo zoccolo di U2, rispettando il verso corretto della la tacca di riferimento. Saldare quindi i due condensatori elettrolitici, le due morsettiere, i connettori per le schede a relè e ICSP, il pulsante. Infine potete saldare la decodifica U1: un consiglio

per i meno esperti è quello di non scaldarla troppo perché potrebbe soffrirne. A questo punto montate il microprocessore e siete pronti per la prima accensione.

PRIMA ACCENSIONE E PROGRAMMAZIONE

In questa fase occorre disporre di un alimentatore da laboratorio regolato a 5V (+/- 0,1V) meglio se limitato in corrente a 200 mA, di un PC con installato un qualsiasi programmatore per PIC (noi abbiamo utilizzato il vecchio glorioso e sempre funzionante PicKit2) ed il file HEX scaricabile dal sito della Rivista.

Serve anche un telecomando, ad esempio il TX-4M-HCS già citato nelle righe precedenti.

Alimentate la scheda e verificate che l'assorbimen-

to a 5 V sia pressoché zero mA (sul nostro prototipo abbiamo verificato circa 4 mA a microprocessore non ancora programmato). Programmate il microcontrollore con l'HEX scaricato dal sito e non prestate attenzione ai led che lampeggeranno a caso durante tutta questa fase; non dimentichiamo infatti che le linee di programmazione sono utilizzate sia per la ricezione dalla decodifica sia per comandare le uscite relè.

Al termine della programmazione entro due secondi il led BIT3 emetterà un certo numero di lampeggi (da 1 a 4) ad indicare il modo di funzionamento predisposto nell'apparato secondo lo specchietto seguente:

- 1 lampeggio: modo treni con telecomando a 4 pulsanti
- 2 lampeggi: modo treni con telecomando a 12 pulsanti
- 3 lampeggi: modo 12 canali con telecomando a 4 pulsanti
- 4 lampeggi: modo 12 canali con telecomando a 12 pulsanti.

L'impostazione predefinita scelta da noi è quella dei tre lampeggi, ossia 12 canali con telecomando a 4 pulsanti, ma nelle prossime pagine vedrete come possa essere variata semplicemente agendo sul telecomando.

UTILIZZO IN PRATICA

Giunti a questo punto, avete a disposizione la vostra scheda montata e funzionante per cui non vi rimane che memorizzare un telecomando ed iniziare ad adoperarla.

Apprendimento del telecomando

Per avviare la procedura è sufficiente premere per un breve istante il pulsante della scheda: il led verde Learn inizia a lampeggiare velocemente indicando che la ricevente è pronta a ricevere "un nuovo telecomando". Premete quindi entro 10 secondi un pulsante del telecomando che state memorizzando, facendo attenzione che il led verde rimanga acceso fisso 2 secondi per poi spegnersi, a conferma che il telecomando è stato memorizzato.

Da ora ogni qualvolta premerete un pulsante del telecomando, vedrete sui led rossi bit0+bit3 il codice binario relativo al tasto premuto e i led delle uscite accendersi in funzione del tasto stesso.

Qualora voleste cancellare tutti i telecomandi appresi, la procedura è molto semplice:

1. si preme per un istante il pulsante e il led verde inizia a lampeggiare;



Fig. 1
telecomando
a 12 pulsanti
modello Aurel
TX-12CH-
HCS.



Fig. 2
Telecomando
a 4 pulsanti
modello Aurel
TX-4M-HCS.

2. si preme nuovamente e si mantiene premuto il pulsante: il led verde si accende a luce fissa;
3. dopo 5 secondi il led verde si spegne; rilasciare il pulsante;
4. il led verde emette 5 lampeggi ad indicare che ora la memoria è vuota e nessun telecomando è memorizzato.

Impostazione del modo di funzionamento

È una procedura molto semplice: per prima cosa occorre aver memorizzato uno o entrambi i telecomandi sulla ricevente secondo la procedura già vista; a questo punto occorre spegnere tutto il circuito, premere uno dei pulsanti secondo quanto riportato nella **Tabella 1**, riaccendere il circuito e mantenere premuto il pulsante del telecomando per almeno 8 secondi.

Durante la fase di accensione si vedrà il led verde learn emettere un lampeggio, quindi si accenderà uno dei led rossi bit0+bit3, secondo la corrispondenza con il tasto premuto.

Rilasciare ora il pulsante del telecomando e verificare che il led rosso bit3 risponda entro 2 secondi con da 1 a 4 lampeggi a conferma del nuovo modo operativo impostato.

PULSANTE TELECOMANDO	LED ROSSO ACCESO	MODO OPERATIVO	RISPOSTA LAMPEGGI LED BIT3
2	bit1	modo treni con telecomando a 4 pulsanti	1
4	bit2	modo treni con telecomando a 12 pulsanti	2
8	bit3	modo 12 canali con telecomando a 4 pulsanti	3
1	bit0	modo 12 canali con telecomando a 12 pulsanti.	4

Tabella 1
Funzione dei pulsanti durante l'impostazione del modo operativo. Le Fig. 1 e 2 mostrano le corrispondenze dei pulsanti nei due telecomandi.

Modo 12 canali con telecomando a 4 pulsanti

Questa condizione pensiamo sia quella potenzialmente più utilizzata poiché consente di associare ad ogni tasto del telecomando l'azionamento di tre relè distinti, arrivando pertanto a $3 \times 4 = 12$ uscite, comandate tutte in modo monostabile per la durata di 1 secondo.

Facendo riferimento alla foto del telecomando a 4 canali scelto vista in precedenza (**Fig. 2**), la corrispondenza tasto / uscita relè è quella riportata in **Tabella 2**.

Come da tabella, il sistema controlla, oltre al tasto premuto, anche per quanto tempo rimane premuto, pertanto pressioni di durata massima di 1 secondo (molto brevi) attiveranno le uscite da 1 a 4, pressioni superiori a 3 secondi attiveranno le uscite da 9 a 12. Tempi intermedi (da 1 a 3 secondi) attiveranno le uscite da 5 a 8.

Quindi per logica si potrebbero assegnare alle pressioni veloci (uscite da 1 a 4) i comandi degli apparati utilizzati più frequentemente quali cancello, basculante, luce esterna, cancelletto; a quelle intermedie (uscite da 5 a 8) operazioni meno ricorsive e magari soggette ad automatismi propri, come ad esempio forzatura irrigazione manuale, luci esterne aggiuntive, eccetera, ed infine alle pressioni lunghe (uscite da 9 a 12) operazioni molto specifiche e personali, quali ad esempio l'attivazione o disattivazione della centrale anti-intrusione, piuttosto che l'accensione/spegnimento del riscaldamento. Come vedete, il sistema è piuttosto versatile grazie alla scelta del controllo "tasto premuto per quanto tempo".

Poiché all'inizio si dovrà fare un po' l'abitudine

alle pressioni più lunghe, un trucchetto semplice ma di aiuto che vi possiamo suggerire consiste nel contare mentalmente partendo da "milleuno" mentre pigiate un tasto, dato che per dire la parola "milleuno" occorre circa 1 secondo. Va da se che se contiamo mentalmente "milleuno, milledue, millette" avremo tenuto premuto il pulsante del telecomando per circa 3 secondi e quindi verrà attivato uno dei relè da 5 a 8.

Modo 12 canali con telecomando a 12 pulsanti

Questa è certamente la condizione di utilizzo più facile, poiché ad ogni singolo pulsante corrisponde un singolo canale, senza necessità di differenziare la durata della pressione del tasto (da 0,2 secondi in su vale tutto). L'assegnazione della corrispondenza pulsante / relè non è stata fatta seguendo la sequenza numerica dei tasti bensì la geometria della tastiera del telecomando, che è disposta in 3 colonne da 4 pulsanti ognuna. Pensando di utilizzare il telecomando senza guardarlo, secondo noi risulta più semplice identificare col pollice la riga e la colonna, piuttosto che il singolo pulsante con un proprio riferimento numerico, quindi abbiamo scelto di assegnare alle uscite da 1 a 4 la colonna di sinistra, alle uscite da 5 a 8 la colonna centrale e alle uscite da 9 a 12 la colonna di destra. La **Tabella 3** riporta la corrispondenza tasto/uscita completa. La **Tabella 4**, invece riporta la corrispondenza pulsanti con i due telecomandi previsti.

Modo treni con telecomando a 4 pulsanti

Come premesso all'inizio, questo progetto è nato per gestire in maniera semplice ed automatica un diorama ferroviario schematizzato nella **Fig. 3**, in

Tabella 2
Corrispondenza "tasto-durata" e uscita, con telecomando a 4 pulsanti.

TASTO PREMUTO...	PER ... <1 SECONDO	PER ... 1+3 SECONDI	PER ... >3 SECONDI
2	Uscita 1	Uscita 5	Uscita 9
4	Uscita 2	Uscita 6	Uscita 10
8	Uscita 3	Uscita 7	Uscita 11
1	Uscita 4	Uscita 8	Uscita 12

7: Uscita 1	8: Uscita 5	9: Uscita 9
4: Uscita 2	5: Uscita 6	6: Uscita 10
1: Uscita 3	2: Uscita 7	3: Uscita 11
Shift: Uscita 4	0: Uscita 8	Enter: Uscita 12

Tabella 3
Corrispon-
denza tasto
telecomando
12 canali e
uscita.

TELECOMANDO A 4 PULSANTI	TELECOMANDO A 12 PULSANTI	FUNZIONE
2 breve	7	Attiva per 1 secondo il canale / relè 1
4 breve	4	Attiva per 1 secondo il canale / relè 2
8 breve	1	Attiva per 1 secondo il canale / relè 3
1 breve	Shift	Attiva per 1 secondo il canale / relè 4
2 medio	8	Attiva per 1 secondo il canale / relè 5
4 medio	5	Attiva per 1 secondo il canale / relè 6
8 medio	2	Attiva per 1 secondo il canale / relè 7
1 medio	0	Attiva per 1 secondo il canale / relè 8
2 lungo	9	Attiva per 1 secondo il canale / relè 9
4 lungo	6	Attiva per 1 secondo il canale / relè 10
8 lungo	3	Attiva per 1 secondo il canale / relè 11
1 lungo	Enter	Attiva per 1 secondo il canale / relè 12

Tabella 4
Corrispon-
denza
pulsanti
telecomandi
in modo
12 canali.

cui sono presenti quattro binari di stazionamento (Avvio 1 + Avvio 4, in blu) con relativo semaforo (Semaforo 1 + Semaforo 4), due coppie di scambi (scambio 1 e scambio 2) che selezionano binario 1 / 2 il primo e binario 3 / 4 il secondo. Si notino, inoltre, la sezione di percorso sempre alimentata (in rosso), il ritorno di massa (in nero), e gli automatismi di attuazione degli scambi (in verde). La condizione tipica di utilizzo di questo schema di percorso prevede

- tutti i semafori predisposti al rosso
- tutti gli avvii (in blu) isolati dal resto del circuito e non alimentati (notare che un avvio è sia avvio che arrivo, per cui quando un treno ci si trova sopra e questo viene attivato, il treno parte, mentre quando il treno è in arrivo e vi transita sopra, trovandolo non alimentato, vi si ferma).
- tutti gli scambi commutati su un binario noto (esempio 1 e 4, per chiudere il giro esterno)

- la linea sempre attiva alimentata alla tensione necessaria per i locomotori (tipicamente 12 V dc ma non è una regola).
- non più di due convogli (anche se con un certo grado di attenzione e di esperienza potrebbero diventare quattro) con motrice posizionata sopra le tratte di avvio in modo da non essere alimentata e pertanto ferma.



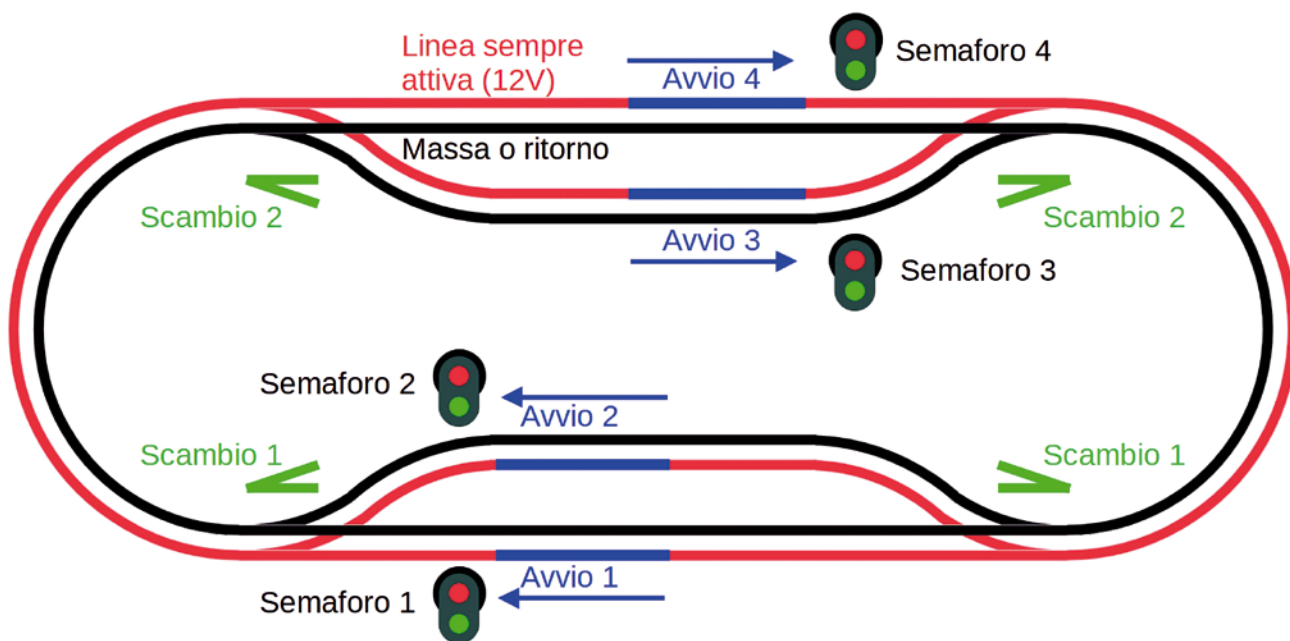


Fig. 3
schema di principio del percorso ferroviario previsto per l'applicazione in modo treni.

Quale che siano la sua posizione e il convoglio che si desidera muovere, la procedura di comando è sempre la stessa (in linea di massima, ovviamente, poiché anche in questo caso non vi sono regole se non quelle previste dal proprio gusto nel gioco) ossia:

1. si cerca il binario di destinazione libero e si commuta lo scambio su di esso;
2. si decide se il convoglio ci si deve fermare (arrivo/avvio non alimentato e semaforo rosso) oppure proseguire (avvio attivo e semaforo verde);
3. si commuta lo scambio di partenza sul lato del convoglio da muovere;
4. si abilita il semaforo relativo al verde;
5. si comanda il binario di avvio fornendogli tensione per il tempo sufficiente a far partire il convoglio quindi lo si disattiva nuovamente;
6. si decide se il convoglio che tornerà su questo tratto dovrà proseguire (binario attivo e semaforo verde) oppure fermarsi (binario non attivo e semaforo rosso).

Come potete constatare, la sequenza non è così facile ed immediata e, mentre nei sistemi di controllo digitali moderni tutto ciò può essere configurato ed automatizzato (pagando ovviamente il costo alto degli apparati di controllo e gestione) in una soluzione analogica più semplice, dove talvolta le dimensioni non giustificano l'investimento per un sistema digitale, le sequenze mettono l'utente in condizioni di poter anche sbagliare comandi. Di fatto, una partenza con un semaforo rosso non è

di per se cosa grave, mentre lo potrebbe essere l'arrivo su un binario che è già occupato. Per semplificare la vita, quindi, il ricevitore è stato progettato per poter gestire quattro macroblocchi di comandi automatizzati, quattro sequenze semi automatiche e 12 semi automatiche o completamente manuali. In pratica esso attiva i relè autonomamente rispettando la sequenza descritta poche righe sopra.

Avvio completamente automatico

Quattro tasti di telecomando premuti brevemente (meno di 1 secondo) corrispondono a quattro operazioni automatiche sui binari di avvio. Quando si preme un tasto, l'apparato esegue la seguente sequenza di operazioni:

1. impulso di comando da 0,5 secondi all'automatismo dello scambio, per posizionarlo sul binario del convoglio: viene dato solo se necessario e il sistema ricorda la posizione dello scambio stesso;
2. dopo 1 secondo viene posto il semaforo al verde;
3. dopo 1 secondo viene comandato il tratto di avvio, così da far partire il convoglio;
4. dopo 5 secondi l'avvio viene spento;
5. dopo 1 secondo il semaforo torna al rosso.

Questa sequenza fa sì che il convoglio in partenza dal binario si muova poco dopo il sopraggiungere verde, ma quello in arrivo ci si fermi trovando il rosso. Inoltre il sistema interblocca i binari 1 / 2 e 3 / 4, ossia impedisce che due tratti affacciati sullo

TELECOMANDO A 4 PULSANTI	TELECOMANDO A 12 PULSANTI	FUNZIONE
2 breve	7	Attiva sequenza automatica di partenza per Avvio 1
4 breve	4	Attiva sequenza automatica di partenza per Avvio 2
8 breve	1	Attiva sequenza automatica di partenza per Avvio 3
1 breve	Shift	Attiva sequenza automatica di partenza per Avvio 4
2 medio	8	Attiva sequenza binario passante Avvio 1
4 medio	5	Attiva sequenza binario passante Avvio 1
8 medio	2	Attiva sequenza binario passante Avvio 1
1 medio	0	Attiva sequenza binario passante Avvio 1
2 lungo	9	Porta a Off tutti gli avvii e al rosso tutti i binari
4 lungo	6	Attiva procedura di verifica semi automatica. Vedere Tabella 6.
8 lungo	3	Attiva procedura di verifica manuale. Vedere Tabella 7.
1 lungo	Enter	Riparte da principio, con tutto off, tutti rossi, Scambi su 1 e 4.

Tabella 5
Corrispon-
denza pulsanti
telecomandi
in modo treni.

stesso scambio siano attivi contemporaneamente, quindi se viene attivato il binario 1, il 2 viene spento insieme al suo semaforo che passa al rosso. La stessa cosa avviene viceversa e con i binari 3 e 4.

Avvio in modalità passante

Quattro tasti di telecomando premuti per un tempo medio (da 1 a 3 secondi) corrispondono a quattro operazioni automatiche sui binari di avvio. A tasto premuto, viene eseguita la seguente sequenza di operazioni:

1. impulso di comando da 0,5 secondi (se necessario) all'automatismo dello scambio, per posizionarlo sul binario dove si trova il convoglio;
2. dopo 1 secondo viene posto il semaforo al verde
3. dopo 1 secondo viene comandato e mantenuto attivo il tratto di avvio, così da far partire il convoglio.

Questa sequenza fa sì che il convoglio in partenza si muova al verde, e quello in arrivo ci transiti sopra trovando anch'esso il verde; permette inoltre di creare un percorso completo sempre attivo in cui il convoglio continui a girare. Ad esempio, dando il comando in modalità passante agli avvii 1 e 4, si abilita il percorso esterno per far girare sempre il convoglio in movimento. Anche in questo caso il sistema interblocca i binari 1 / 2 e 3 / 4 per cui solamente un binario per scambio potrà essere attivo.

Volendo fermare il convoglio, si può dare un comando automatico (una pressione breve) al binario su cui si vuole che si fermi, oppure inviare il comando di stop, facente parte dei comandi speciali che vedremo tra poco.

Comandi speciali

Quattro tasti di telecomando premuti per un tempo lungo (oltre i 5 secondi) corrispondono quattro funzioni speciali.

Tasto 2: determina il passaggio al rosso di tutti i semafori e a spento di tutte le tratte di avvio. Quindi il convoglio si fermerà al primo tratto di avvio spento che troverà. Gli scambi non vengono comandati.

Tasto 1: ripristina la condizione di partenza, ossia scambi su binari 1 e 4, semafori tutti rossi e avvii tutti a riposo.

Tasto 4 e 8: avviano le modalità di collaudo semi automatica (4) e manuale (8), scopo delle quali non è quello di far muovere i convogli, ma solo fornire delle condizioni di test note con cui verificare a livello strumentale che tutti i collegamenti elettrici siano a posto. Abbiamo riassunto tutte le funzioni nelle **Tabelle 6 e 7**.

Modo treni con telecomando a 12 pulsanti

Funziona in modo identico a quello a 4 pulsanti, con la sola differenza che con 12 tasti non occorre



Tabella 6

Corrispondenza pulsanti telecomandi in modo treni, verifica semi automatica. Il sistema esce automaticamente da questa modalità dopo 12 secondi di inattività sul telecomando.

TELECOMANDO A 4 PULSANTI	TELECOMANDO A 12 PULSANTI	FUNZIONE
2 breve	7	Attiva semaforo 1 per 1 secondo
4 breve	4	Attiva semaforo 2 per 1 secondo
8 breve	1	Attiva semaforo 3 per 1 secondo
1 breve	Shift	Attiva semaforo 4 per 1 secondo
2 medio	8	Attiva avvio 1 per 2 secondi
4 medio	5	Attiva avvio 2 per 2 secondi
8 medio	2	Attiva avvio 3 per 2 secondi
1 medio	0	Attiva avvio 4 per 2 secondi
2 lungo	9	Attiva scambio 1 binario 1 per 1 secondo
4 lungo	6	Attiva scambio 1 binario 2 per 1 secondo
8 lungo	3	Attiva scambio 2 binario 3 per 1 secondo
1 lungo	Enter	Attiva scambio 2 binario 4 per 1 secondo

Tabella 7

Corrispondenza pulsanti telecomandi in modo treni, verifica manuale. Il sistema esce automaticamente da questa modalità dopo 120 secondi di inattività sul telecomando.

TELECOMANDO A 4 PULSANTI	TELECOMANDO A 12 PULSANTI	FUNZIONE
2 breve	7	Attiva / disattiva semaforo 1
4 breve	4	Attiva / disattiva semaforo 2
8 breve	1	Attiva / disattiva semaforo 3
1 breve	Shift	Attiva / disattiva semaforo 4
2 medio	8	Attiva / disattiva avvio 1
4 medio	5	Attiva / disattiva avvio 2
8 medio	2	Attiva / disattiva avvio 3
1 medio	0	Attiva / disattiva avvio 4
2 lungo	9	Attiva scambio 1 binario 1 per 1 secondo
4 lungo	6	Attiva scambio 1 binario 2 per 1 secondo
8 lungo	3	Attiva scambio 2 binario 3 per 1 secondo
1 lungo	Enter	Attiva scambio 2 binario 4 per 1 secondo

fare le differenziazioni sul tempo premuto. Nella **Tabella 5** abbiamo riportato la corrispondenza delle operazioni effettuate con telecomando a 4 tasti e a 12 tasti.

SCHEMA DI PRINCIPIO PER L'UTILIZZO IN MODO TRENI

Arrivati a questo punto, si è già consapevoli di come l'utilizzo tradizionale per comando remoto di cancelli, basculanti... sia facile ed intuitivo: basta infatti identificare l'ingresso di comando sulla

scheda elettronica dell'apparato di destinazione (che tipicamente necessita solo di un contatto normalmente aperto, come un pulsante) e collegare ad esso tramite due fili il contatto normalmente aperto di una delle uscite a relè.

In ambito fermodellistico, però, la cosa è un po' più complessa: gli scambi possono essere del tipo a 3 fili a comando di direzione individuale, oppure a 2 fili ad inversione di polarità; i semafori possono essere a lampadina o a led, a 2 o 3 fili; le tratte di avvio devono essere isolate e distinte dalle tratte

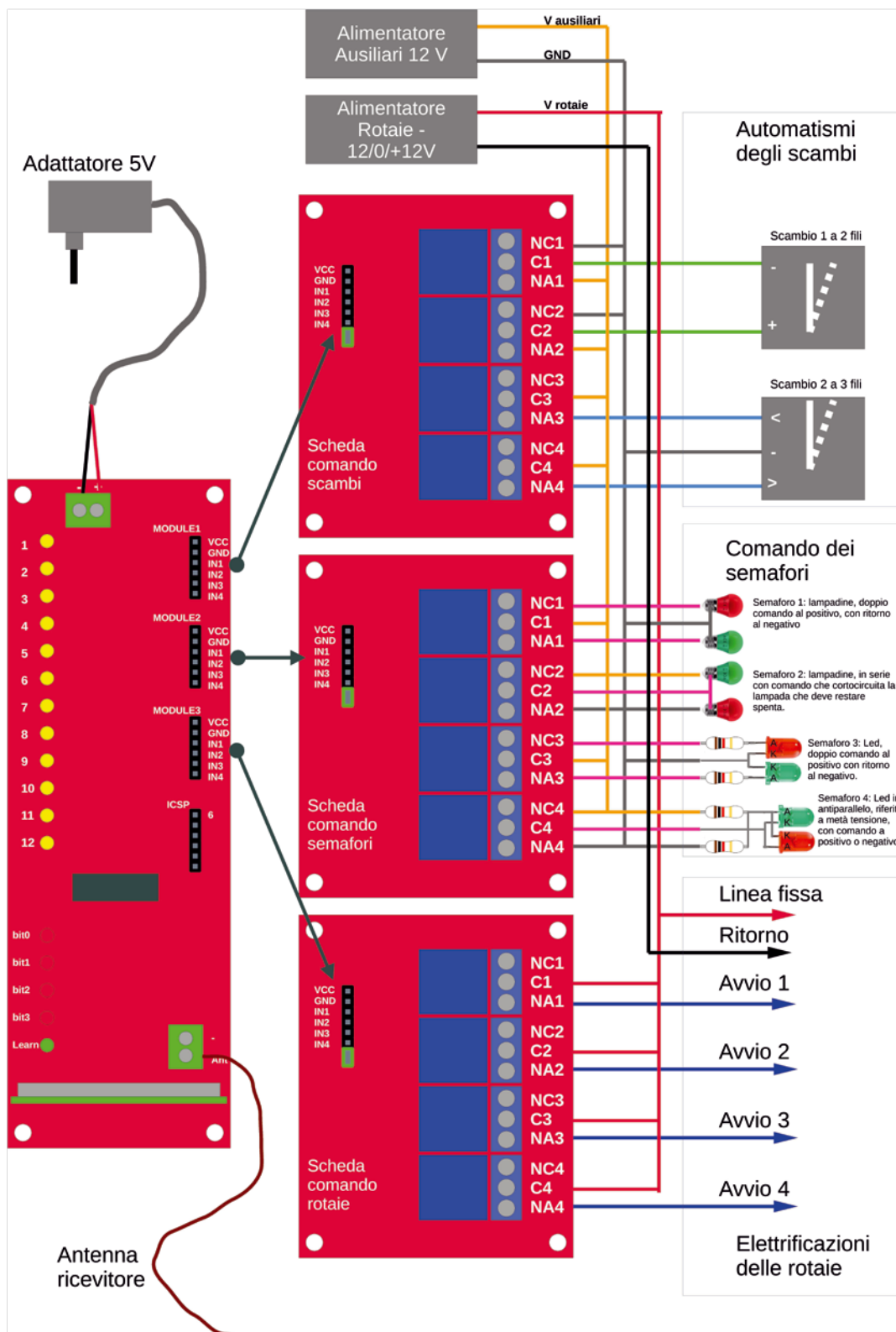


Fig. 4
Schema elettrico di principio per l'utilizzo in modo treni.

Si tratta di un modulo dotato di 4 relè che consente di attivare/disattivare il carico ad essi collegato, con tensione fino a 250V e corrente fino a 10A. Dispone di ingressi optoisolati e LED di stato per ogni ingresso, più un led di segnalazione di presenza alimentazione.

SPECIFICHE TECNICHE

- Alimentazione: 5 Vdc
- Numero relè: 4
- Contatti di uscita: N.A. – Comune – N.C.
- Portata contatti: 10A 250Vac – 10A 30Vdc
- Tensione attivazione: 2,5 – 5 Vdc
- Dimensioni (mm): 77,50x55x 19



ad alimentazione continua, eccetera. Inoltre possono essere utilizzate differenti metodologie di cablaggio, per cui creare una casistica completa risulta alquanto complesso. Abbiamo così riassunto in **fig. 4** alcune delle situazioni più comuni che possono essere ritrovate nei sistemi fermodellistici completamente manuali.

Vediamole una per volta, tenendo presente che il sistema necessita di tre alimentazioni distinte:

- Adattatore a 5Vdc per alimentare la sezione elettronica
- Alimentatore a 12 V fisso, dedicato all'attuazione di scambi e semafori
- Alimentatore 0÷12V reversibile, per uso fermodellistico, dedicato alla movimentazione dei convogli

Si noti, inoltre, che le masse dei due alimentatori a 12 V non sono collegate tra loro (ma qualora lo fossero, non ci sarebbero controindicazioni, a patto di avere un negativo di sezione adeguata), così come la sezione a 5 Vdc è isolata galvanicamente dal resto del circuito ferroviario.

Scambio comandato a 2 fili

Questo tipo di scambio a comando elettromeccanico, il cui schema elettrico di principio è visibile in **Fig. 5e**, dispone di una sola bobina che svolge la funzione di elettromagnete e prevede l'applicazione di un comando elettrico di breve durata a

polarità diritta o invertita a seconda che si voglia deviarlo su un binario o sull'altro. In pratica, applicando ai morsetti + e – un impulso elettrico con positivo sul + otterremo lo spostamento del derivatore meccanico in senso orario. Analogamente, applicando l'impulso positivo sull'ingresso –, otterremo lo spostamento del derivatore il posizione antioraria. Nello schema di **fig. 4**, lo scambio a 2 fili è gestito dalle uscite 1 e 2 del modulo a relè 1, in cui i contatti NC sono elettricamente collegati al negativo degli ausiliari, e i comandi NA sono al positivo. I due comuni C1 e C2 vanno agli ingressi + e – dello scambio che, pertanto, in condizione di riposo risultano sempre elettricamente collegati al negativo. Comandando il relè 1 viene dato un positivo all'ingresso – dello scambio, mentre se si comanda il relè 2 il positivo viene dato all'ingresso + dello scambio, ottenendo di fatto lo spostamento da una parte o dall'altra.

Scambio comandato a 3 fili

Questo tipo di scambio (**Fig. 5f**), a differenza del precedente, dispone di due bobine distinte, con un capo in comune, normalmente collegato al negativo, e due positivi distinti. Dando il comando a un positivo o all'altro (la contemporaneità non è prevista) si ottiene lo spostamento voluto. Il cablaggio è quello associato alle uscite 3 e 4 (**Fig. 4**) della scheda comando scambi. Di fatto il comune delle bobine è connesso al negativo, i contatti normalmente aperti dei relè sono

Tabella 8
Decodifica telecomando a 4 pulsanti (TLC4), ricodifica a 12 pulsanti (TLC12) e relativa costante nel firmware.

TASTO PREMUTO... (FIG 1)	<1 SECONDO	1÷3 SECONDI	>3 SECONDI
TLC4, tasto 2	TLC12, tasto 7, T_7	TLC12, tasto 8, T_8	TLC12, tasto 9, T_9
TLC4, tasto 4	TLC12, tasto 4, T_4	TLC12, tasto 5, T_5	TLC12, tasto 6, T_6
TLC4, tasto 8	TLC12, tasto 1, T_1	TLC12, tasto 2, T_2	TLC12, tasto 3, T_3
TLC4, tasto 1	TLC12, tasto SHIFT, T_SH	TLC12, tasto 0, T_0	TLC12, tasto ENTER, T_EN

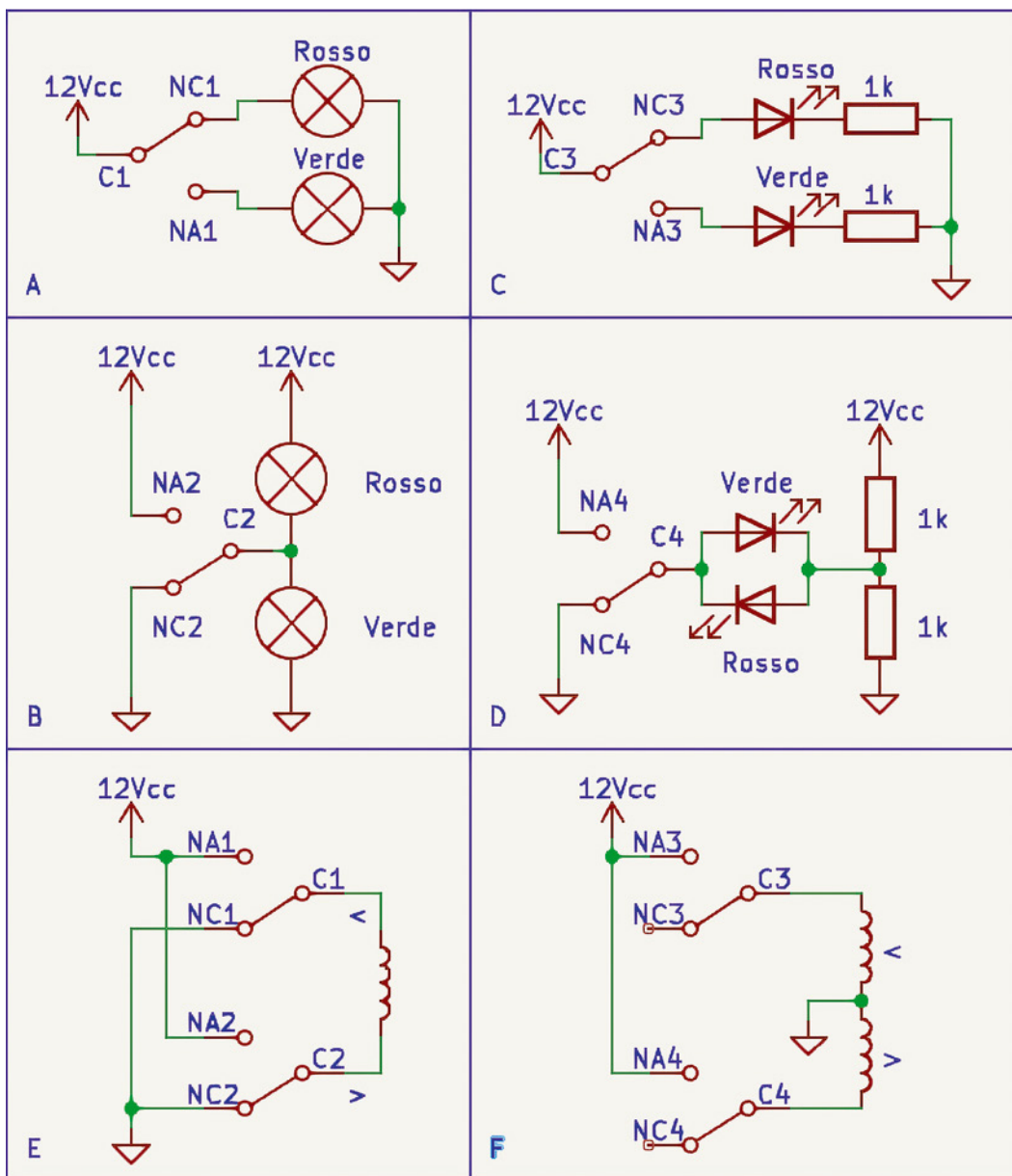


Fig. 5
Schemi comando scambi e semafori.

collegati al positivo ausiliari, mentre i comuni dei contatti vanno ai due comandi di direzione dello scambio.

Comando semaforico a 3 fili, con lampadine

Questo tipo di semaforo (**Fig. 5a**) dispone di tre fili: un comune e due comandi individuali del rosso o del verde. La condizione di comando più semplice è quella realizzata (**Fig. 4**) dal comando 1 della scheda comando semafori, in cui il comune del semaforo è al negativo mentre i due comandi individuali fanno capo al normalmente chiuso e al normalmente aperto del relè. Il comu-

ne del relè a sua volta va al positivo ausiliari. Di fatto rimane sempre accesa la lampadina associata al contatto normalmente chiuso (tipicamente il rosso) e pertanto, comandando il relè 1, si otterrà lo spegnimento del rosso e l'accensione del verde.

È possibile utilizzare anche la gestione realizzata dall'uscita 2 del comando semafori (schema in **Fig. 5b**), in cui il relè cortocircuita una o l'altra lampadina. Funziona ma è poco utilizzata perché più complessa della precedente, pur offrendo il vantaggio di comandare entrambe le luci con un solo filo.

Listato 1

```

'*****
' Legge la ricevente
LETTURA_RADIO:
  SELECT case TLC_SEL      ' Verifica modo e telecomando.
  case P12_TR, P12_12CH    ' TLC e' 12 pulsanti.
    gosub LEGGE_RXD1      ' legge una volta.
    gosub DELAY_50ms      ' Attende.
    gosub LEGGE_RXD2      ' Rilegge.
    IF RXD1 = RXD2 then    ' T_2 letture uguali
      TASTO = RX_OFF-RXD1 ' elabora il tasto
    else
      TASTO = T_NUL      ' altrimenti nessun tasto
    return
  endif
  gosub RADIO_A_RIPOSO    'attende tasti a riposo
  case P4_TR, P4_12CH
    gosub LEGGE_RXD1      ' col 4 pulsanti legge il tempo premuto.
    gosub DELAY_50ms      ' Attende 50 ms
    gosub LEGGE_RXD2      ' e rilegge il telecomando.
    IF RXD2 <> rxd1 THEN RETURN ' letture non coincidono: esce
    if RXD2 = RX_OFF then  ' Letture coincidenti e pari a nessun tasto
      TASTO = T_NUL      ' premuto: azzera contatore, tasto ed esce.
      RX_CNT = 0
    return
  endif
  WHILE RXD2 <> RX_OFF    ' rilegge in continuazione per sapere
    RX_CNT = RX_CNT+1    ' da quanto tempoo è premuto.
    gosub DELAY_50ms
    gosub LEGGE_RXD2
  WEND
  if CHECK_MODE = 1 then write 6,RX_CNT ' solo per debug
  select case RX_CNT
    case is <20
      select case RXD1
        case 13
          TASTO = T_7      ' tasto 2 premuto, bit 1 a 0
        case 11
          TASTO = T_4      ' corrisponde a 7 nel 12 pulsanti
        case 7
          TASTO = T_1      ' tasto 4 premuto, bit 2 a 0
        case 14
          TASTO = T_1      ' corrisponde a 4 nel 12 pulsanti
        case 14
          TASTO = T_1      ' tasto 8 premuto, bit 3 a 0
        case 14
          TASTO = T_1      ' corrisponde a 1 nel 12 pulsanti
        case 14
          TASTO = T_1      ' tasto 1 premuto, bit 0 a 0
        case 14
          TASTO = T_SH     ' corrisponde a Shift nel 12 pulsanti
      end select
    case is >60
      select case RXD1
        case 13
          TASTO = T_9      ' pressione oltre i 3 secondi
        case 11
          TASTO = T_9      ' tasto 2 premuto, bit 1 a 0
        case 7
          TASTO = T_6      ' corrisponde a 9 nel 12 pulsanti
        case 14
          TASTO = T_6      ' tasto 4 premuto, bit 2 a 0
        case 14
          TASTO = T_6      ' corrisponde a 6 nel 12 pulsanti
        case 14
          TASTO = T_3      ' tasto 8 premuto, bit 3 a 0
        case 14
          TASTO = T_3      ' corrisponde a 3 nel 12 pulsanti
        case 14
          TASTO = T_3      ' tasto 1 premuto, bit 0 a 0
        case 14
          TASTO = T_EN     ' corrisponde a ENTER nel 12 pulsanti
      end select
    case else
      select case RXD1
        case 13
          TASTO = T_8      ' pressione tra 1 e 3 secondi
        case 11
          TASTO = T_8      ' tasto 2 premuto, bit 1 a 0
        case 7
          TASTO = T_5      ' corrisponde a 8 nel 12 pulsanti
        case 14
          TASTO = T_5      ' tasto 4 premuto, bit 2 a 0
        case 14
          TASTO = T_5      ' corrisponde a 5 nel 12 pulsanti
        case 14
          TASTO = T_2      ' tasto 8 premuto, bit 3 a 0
        case 14
          TASTO = T_2      ' corrisponde a 2 nel 12 pulsanti
        case 14
          TASTO = T_0      ' tasto 1 premuto, bit 0 a 0
        case 14
          TASTO = T_0      ' corrisponde a 0 nel 12 pulsanti
      end select
    END SELECT
    if CHECK_MODE = 1 then WRITE 7,TASTO ' solo debug.
    rx_cnt = 0 ' azzera contatore di tasto premuto.
  end select
  return
'*****
' attesa radio a riposo, ossia nessun tasto telecomando premuto.
RADIO_A_RIPOSO:
  gosub LEGGE_RXD1
  while RXD1 <>15
    gosub LEGGE_RXD1
  wend
  return
'*****
' Lettura bitwise della radio
' a conteggiare bitwise si risparmiano 58 byte programma
LEGGE_RXD1
  RXD1=0
  RXD1.0=bit0
  RXD1.1=bit1
  RXD1.2=bit2
  RXD1.3=bit3
  return
LEGGE_RXD2
  RXD2=0
  RXD2.0=bit0
  RXD2.1=bit1
  RXD2.2=bit2
  RXD2.3=bit3
  return

```

Comando semaforico a 3 fili, con LED

Rappresentato dallo schema in **fig. 5c**, funzionalmente è identico al comando a tre fili a lampadine, ed è realizzato dall'uscita 3 della scheda comando semafori. Il cablaggio delle uscite 1 e 3, è identico.

Comando semaforico a 1 filo, con LED

Questa soluzione è schematizzata nella **fig. 5d**: non è realizzata nei semafori acquistabili di serie, ma si presta particolarmente per la realizzazione artigianale. Visibile sull'uscita 4 della scheda semafori, il sistema prevede due resistenze in serie, collegate tra positivo e negativo degli ausiliari, così da ottenere un "riferimento o nodo" intermedio a metà tensione. I due led rosso e verde sono collegati in antiparallelo (il catodo di ognuno è collegato all'anodo dell'altro) e fanno capo da una parte al nodo a metà tensione, e dall'altro al filo di comando, per cui si possono ottenere tre condizioni di comando distinte:

- filo scollegato: i led sono spenti
- filo al positivo di ausiliari: si accende un led
- filo al negativo di ausiliari: si accende l'altro led.

Va da sé che il filo di comando è collegato al comune del relè, mentre il normalmente aperto e il normalmente chiuso sono elettricamente connessi al positivo uno e al negativo l'altro.

Elettrificazione delle rotaie

Questa è la parte più facile di tutto il cablaggio. Sempre facendo riferimento allo schema di principio di **fig. 4**, la traccia nera rappresenta il negativo di rotaia, il rosso è il positivo sempre presente, che è collegato alle tratte di binario sempre alimentate e ai comuni dei quattro relè di comando di avvio. Ogni contatto normalmente aperto degli stessi relè va ad alimentare la propria tratta di avvio, che deve essere isolata dal resto del circuito ferroviario attraverso degli isolatori appositi, in vendita nei negozi specializzati ma anche online, che si interpongono nelle giunte delle rotaie e che gergalmente vengono anche chiamati "scarpette".

NOTE CONCLUSIVE

Qui giunti, non ci sono molte cose da aggiungere: sia che vi occorra un controllo remoto, sia che vogliate giocare con i trenini, in queste pagine vi abbiamo presentato una soluzione facile ed economica valida in entrambi i casi. Tutto il materiale per la realizzazione di questo dispositivo può essere facilmente reperito presso il negozio online della Futura Elettronica. 

La parte più importante di tutto il sorgente è rappresentata dalla lettura della radio e dalla decodifica e ricodifica del tasto letto (Listato 1). Tutto il resto del codice, alla fine, consiste in semplici sequenze del tipo "IF... THEN..." o "SELECT CASE". La lettura viene eseguita chiamando la subroutine LETTURA_RADIO, la quale sfrutta altre subroutines di servizio, in particolare LEGGE_RXD1 / LEGGE_RXD2, dove vengono memorizzate due letture successive distanti "DELAY_50ms" e RADIO_A_RIPOSO, che verifica che nessun tasto sia più premuto. Se le letture RXD1 e RXD2 coincidono, allora viene processata la decodifica/ricodifica.

A lettura valida, si presentano due casistiche: telecomando a 12 pulsanti in modo treni o 12 canali (CASE P12_TR, P12_12CH) o telecomando a 4 pulsanti (CASE P4_TR, P4_12CH) sempre in modo treni o 12 canali.

Con telecomando a 12 pulsanti (CASE P12_TR, P12_12CH) è sufficiente identificare quale pulsante è premuto (ossia decodificarlo) e scriverlo nella variabile TASTO, cui altre parti di programma faranno accesso, quindi uscire dalla subrou-

tine quando la radio comunica che il telecomando è a riposo.

Con telecomando a 4 pulsanti, invece, la cosa è più contorta: abbiamo già detto che in origine il sistema era stato pensato per funzionare con il telecomando a 12 pulsanti, quindi ad ogni pulsante era associato un solo valore decodificato.

Poi però la reperibilità ed il costo di tale telecomando ci hanno obbligato a correre ai ripari, ma purtroppo il codice era già completo e funzionante.

Quindi abbiamo aggiunto del nuovo codice identificato dal "CASE P4_TR, P4_12CH", in cui si eseguono le seguenti operazioni:

1. si decodifica il tasto premuto
2. si conta per quanto tempo è premuto
3. si verifica il rilascio stabile del pulsante e si blocca il tempo
4. secondo una combinazione di 4 codici e 3 tempi, si assegna alla variabile TASTO il valore che avrebbe se fosse utilizzato il telecomando a 12 tasti.
5. a ricodifica effettuata si azzerla la variabile contatempo.

In pratica si decodifica il telecomando a 4 tasti e lo si ricodifica a 12 tasti (Tabella 8).



COSA OCCORRE ?

Codice	Descrizione	Prezzo € IVA inclusa
TX-4M-HCS	Trasmettitore palmare RF HCS a 4 CH	26,50
RX-4MHCS-4B	Ricevitore 4 bit con HCS	17,00
PIC16F88-I/P	Microcontrollore Microchip PIC16F88-I/P	12,00
RELAY4CH	Modulo 4 relè 5 Vdc 10A	11,00
PCBPRODUCTION	Servizio realizzazione PCB	

I restanti componenti sono di facile reperibilità.

IL MATERIALE VA RICHIESTO A:

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

Nuove Telecamere

Full color technology



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

TELECAMERA FULL COLOR 2 MPX - VARIFOCAL



€ 69,00

cod. FR816VAR

TELECAMERA FULL COLOR 2 MPX



€ 52,00

cod. FR816N

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

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

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

TELECAMERA DOME FULL COLOR - 2 MPX

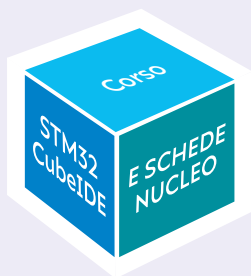
- ✓ Uscita Video (BNC): 4 in 1 - AHD / CVI / TVI / CVBS
- ✓ Sensore di immagine: 1/2.8" CMOS
- ✓ Risoluzione immagine: 2 MPX
- ✓ Pixel effettivi: 1920(H)*1080(V)
- ✓ Sistema TV: PAL
- ✓ Obiettivo: 3,6 mm
- ✓ Otturatore elettronico: Automatico
- ✓ Portata Illuminatore: 20-30 metri
- ✓ Grado di protezione: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

€ 44,00

cod. FR817



Trasformano
la notte 
in  giorno!



4

STM32CubeIDE E SCHEDA NUCLEO: LA POTENZA DELL'EMBEDDED A PORTATA DI MANO

di VALTER SARTORI

Analizziamo la configurazione Pulse Width Modulation del Timer, implementando un progetto per variare la luminosità di un LED e introduciamo la periferica UART per la gestione della trasmissione e ricezione dati. Quarta puntata.

Tra le più importanti modalità d'uso della periferica Timer utilizzate nella maggior parte dei progetti embedded troviamo la Pulse Width Modulation o PWM che genera un'onda quadra di periodo costante, ma con pulsazione modulabile.

La **Fig. 1** mostra tre diverse forme d'onda quadra che condividono lo stesso periodo, e di conseguenza hanno la stessa frequenza.

Tuttavia, ciascuna di queste onde ha un duty cycle

differente: 50%, 20% e 80%. Il duty cycle, espresso in percentuale, rappresenta il rapporto tra il tempo in cui l'onda è attiva (Ton) e la durata totale del periodo dell'onda quadra.

Prendiamo in considerazione lo schema a blocchi di **Fig. 2** e osserviamo che i Timer che andremo a utilizzare sono collegati al bus APB1 per cui prenderemo come riferimento la frequenza 'APB1 Timer clocks' di lavoro per calcolare i parametri di pulsazione del PWM.

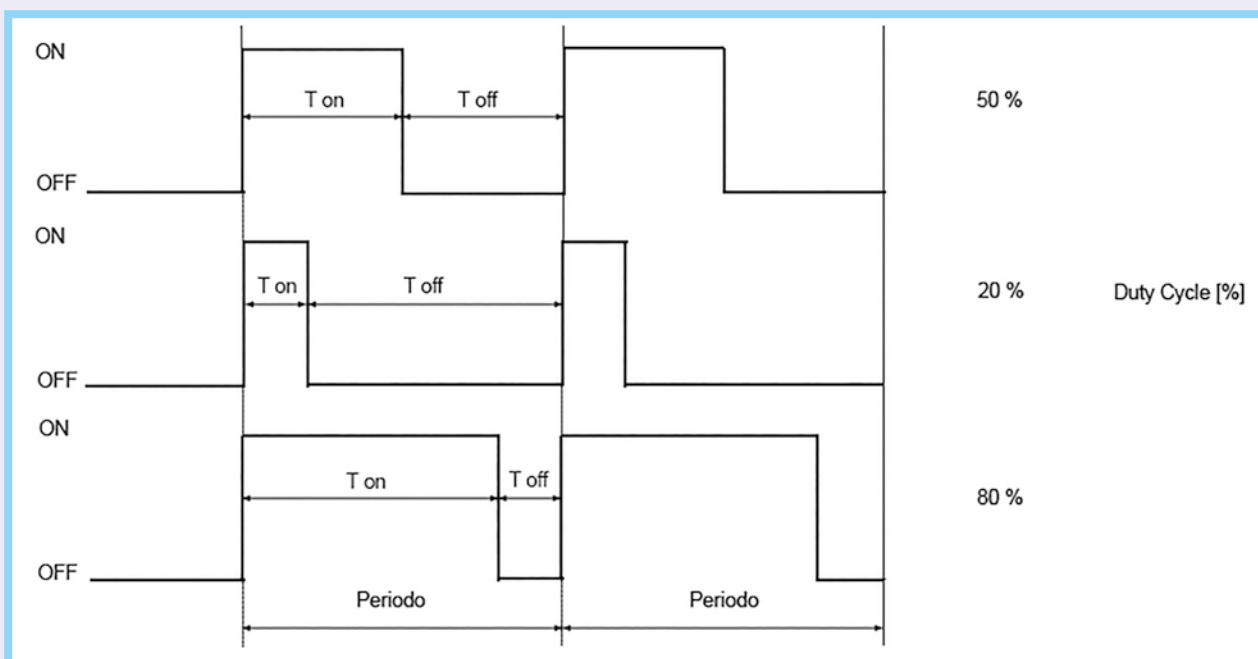


Fig. 1 Analisi delle forme d'onda PWM

Prima di addentrarci nella realizzazione del nostro progetto andremo a determinare gli algoritmi che ci permetteranno di calcolare i parametri per ottenere la modulazione proposta di seguito:

- Periodo onda quadra: 1 ms
- Frequenza: 1 kHz
- Larghezza impulso (T_{on}): 250 μ s
- Duty Cycle: 25%

Partendo dalla formula menzionata nella scorsa puntata e ripresentata qui in basso:

$$Event_{Time} = \frac{(Prescaler + 1)(Period + 1)}{(Timer_{clock})}$$

dove:

- $Event_{Time}$ espresso in s rappresenta la periodicit  dell'evento
- $Timer_{clock}$ espresso in Hz rappresenta la frequenza del bus a cui   collegato il timer
- $Prescaler$ espresso in ticks rappresenta il fattore di divisione
- $Period$ espresso in ticks rappresenta il coefficiente di periodicit  del timer

ricaveremo la formula inversa che ci fornir  il coefficiente da adottare come Counter Period:

$$Period = \frac{Timer_{clock} * Event_{Time}}{Prescaler + 1} - 1$$

Applicheremo la formula inversa per ottenere i coefficienti da inserire nei parametri Prescaler e Counter Period (Auto Reload Register) ottenendo cos  la forma d'onda desiderata. Dopo aver introdotto gli strumenti che ci serviranno per generare la pulsazione scelta, passiamo a realizzare il progetto denominandolo "F401_PWM" e ricordandoci di selezionare il

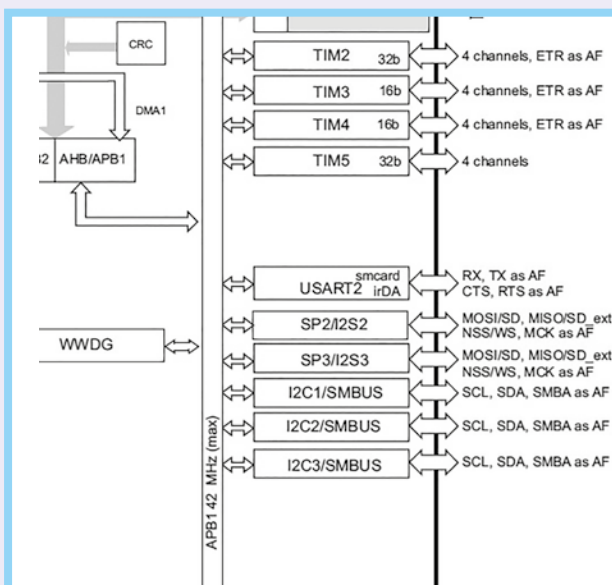


Fig. 2 Schema a blocchi parziale del bus di collegamento dei Timer nel STM32F401

Pinout & Configuration

Clock Configuration

▼ Software Packs

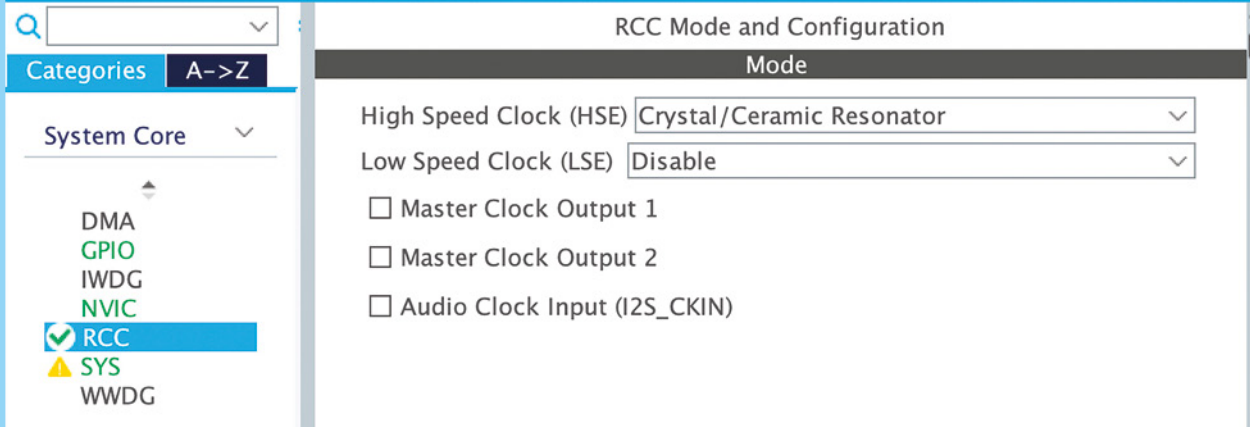


Fig. 3 Configurazione sorgente di clock esterna

tipo di MCU e non la scheda Nucleo come già raccomandato. Quando compare la schermata di STM32CubeMX procediamo come mostrato in **Fig. 3** configurando RCC con sorgente esterna del clock.

Procediamo con la 'Clock Configuration' dove attiveremo i blocchi relativi alla gestione della sorgente esterna di clock. Nella configurazione del clock visibile in **Fig. 4**, possiamo osservare la connessione di HSE con il quarzo esterno e l'abilitazione della linea PLLCLK.

Impostiamo la frequenza massima nel blocco HCLK e premiamo Enter in attesa che la configurazione automatica del sistema si disponga come in **Fig. 4**.

Terminata la 'Clock Configuration' ritorniamo alla tab 'Pinout & Configuration' ed iniziamo a configurare la periferica PWM. Come mostrato in **Fig. 5** espandiamo la categoria 'Timers' e

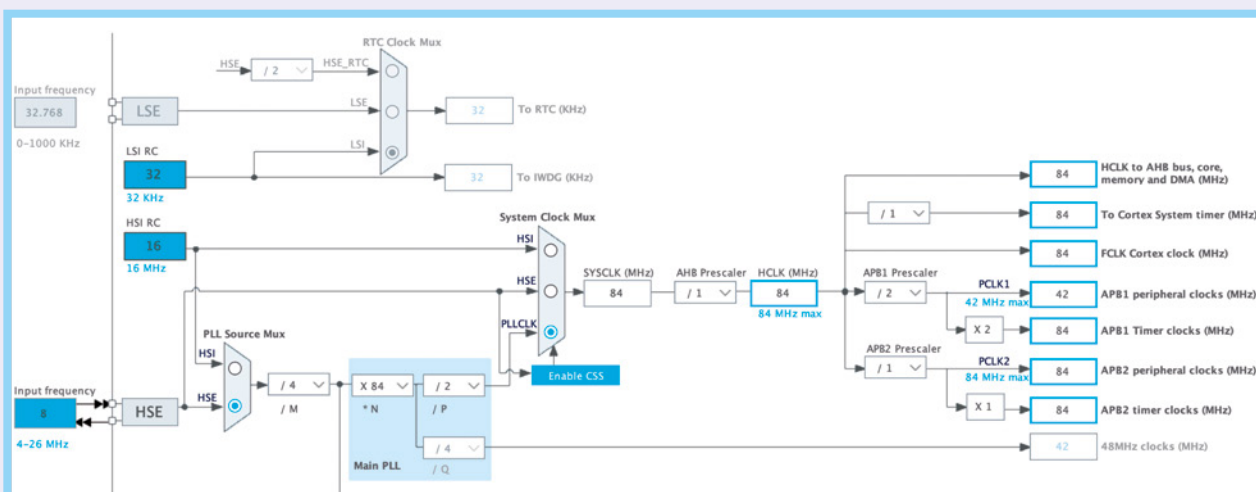
selezioniamo 'TIM3', un timer general purpose con risoluzione a 16 bit (vedere Tabella 3 Puntata 3).

TIM3 è collegato al bus APB1 e la frequenza 'APB1 Timer clocks' di lavoro risulta 84 MHz come è possibile osservare in **Fig. 4**.

Questo dato ci sarà utile per calcolare i parametri di pulsazione del PWM.

Selezionando TIM3, si apriranno le finestre di selezione della Modalità di funzionamento e la configurazione stessa (**Fig. 5**). Nella finestra 'Mode' andremo ad impostare il Clock Source e attiveremo uno dei quattro canali PWM disponibili: 'Channel1'. Per il Clock Source selezioniamo 'Internal Clock', per Channel1 'PWM Generation CH1'.

Imponiamo il parametro Prescaler a 83 (in realtà $84-1$ in quanto il valore 0 viene considerato 1 come fattore matema-



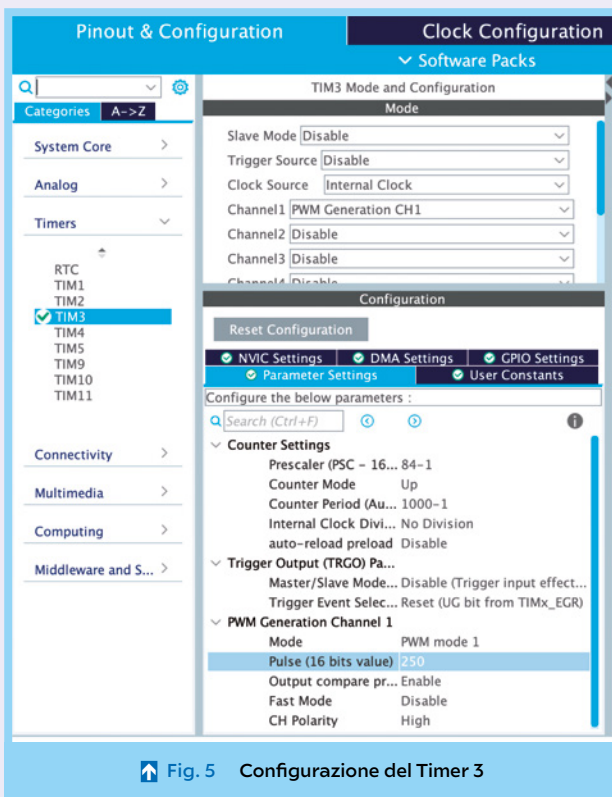


Fig. 5 Configurazione del Timer 3

tico della divisione (vedere puntata 3 paragrafo "Configurazione di un Timer in modalità interrupt")

Ad esempio il calcolo del parametro Period nel nostro caso sarà:

$$999 = \frac{(84000000 * 0.001)}{(83 + 1)} - 1$$

Configurando i parametri come da calcolo otteniamo un'onda quadra a frequenza 1 kHz con periodo 1 ms.

Per ottenere il duty cycle del 25% con una larghezza di impulso di 250 µs occorre applicare la seguente formula:

$$Pulse = \frac{(Duty\%)(Period + 1)}{(100)}$$

che tiene conto del parametro immesso nel registro Counter Period, per cui si ottiene:

$$250 = \frac{(25)(999 + 1)}{(100)}$$

ed è il parametro che andremo ad inserire nel registro Pulse che determina, rivedendo la Fig. 1, la larghezza di T_{on} .

Al termine della configurazione di Timer3 evidenziata in Fig. 5, salviamo il progetto, generando il codice di inizializzazione.

Inseriamo alla riga 92 del file main.c l'istruzione evidenziata in Fig. 6 che attiva l'uscita del segnale PWM.

Colleghiamo la scheda Nucleo F401RE mediante il cavo USB al computer e compiliamo il software scaricandolo nella MCU. Avviando il firmware e collegando l'uscita PA6 ad un analizzatore di segnali otteniamo il segnale di Fig. 7 perfettamente conforme con le specifiche riportate precedentemente.

Questo semplice esempio è servito per addentrarci nel metodo di configurazione della periferica PWM.

Ora affronteremo un progetto più interessante che ha come obiettivo ottenere un segnale PWM dinamico con pulsazioni variabili nel tempo.

Implementeremo un nuovo progetto unendo l'esperienza fatta nella puntata precedente con il convertitore A/D e pilotando un led utilizzando l'uscita PWM precedentemente descritta.

Ruotando il potenziometro collegato al convertitore A/D andremo a variare la luminosità del led.

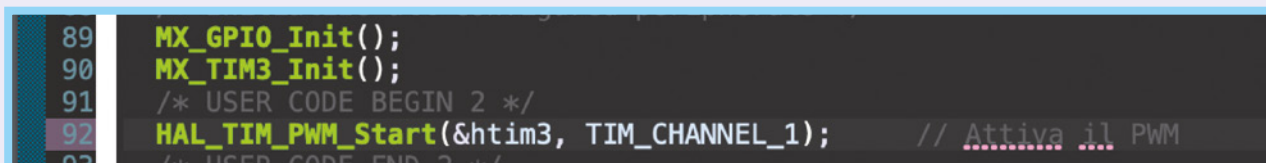


Fig. 6 attivazione software del segnale PWM

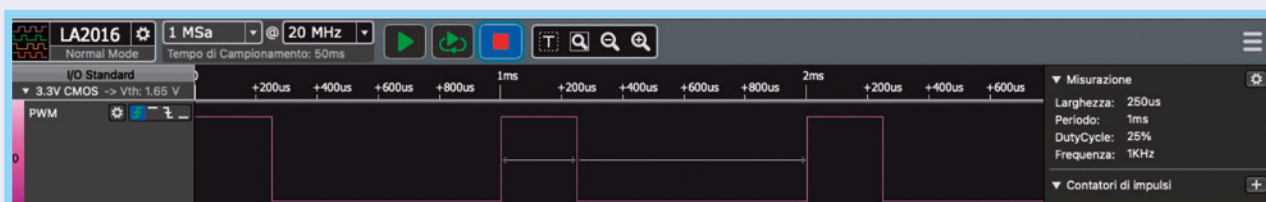


Fig. 7 Analisi del segnale PWM

MODALITÀ PULSE WITH MODULATION DINAMICA

Implementeremo il circuito descritto nelle **Fig. 8 e 9** collegando l'uscita del PWM (PA6) ad un led attraverso una resistenza da 560 Ω . Utilizzeremo un potenziometro collegandolo ad un ingresso A/D per ottenere un segnale variabile utile a far variare il duty cycle della forma d'onda PWM e di conseguenza la luminosità del led.

Andremo a predisporre un nuovo progetto nominandolo F401_PWM_VAR e faremo riferimento alle configurazioni adottate nella precedente puntata, riguardanti la conversione analogico digitale.

Quindi per la parte di configurazione della periferica ADC occorrerà ripercorrere gli stessi passaggi affrontati la volta scorsa.

In **Fig. 10** si possono osservare le configurazioni adottate per la gestione del PWM che ricalcano quelle precedentemente

impostate a parte il registro Pulse dove lasciamo invariato il parametro a zero.

Per quanto riguarda il listato software del file main.c, nelle figure che seguono possiamo valutare le sezioni che sono state oggetto di modifiche e implementazioni software.

In **Fig. 11** vengono evidenziate le variabili utilizzate nella 'main' che verranno utilizzate per il calcolo del duty cycle.

Andremo ad implementare una nuova funzione, come si può osservare nelle **Fig. 12 e 13**, che servirà a calcolare e configurare il registro Pulse che determina la durata dell'impulso. La funzione dutyCycle riceve mediante la variabile 'duty' la durata dell'impulso in percentuale rispetto al periodo della forma d'onda, calcolata in base alla misura del convertitore A/D.

Utilizzando l'ultima formula ricalcoliamo ad ogni aggiornamento della misura del convertitore A/D il parametro Pulse

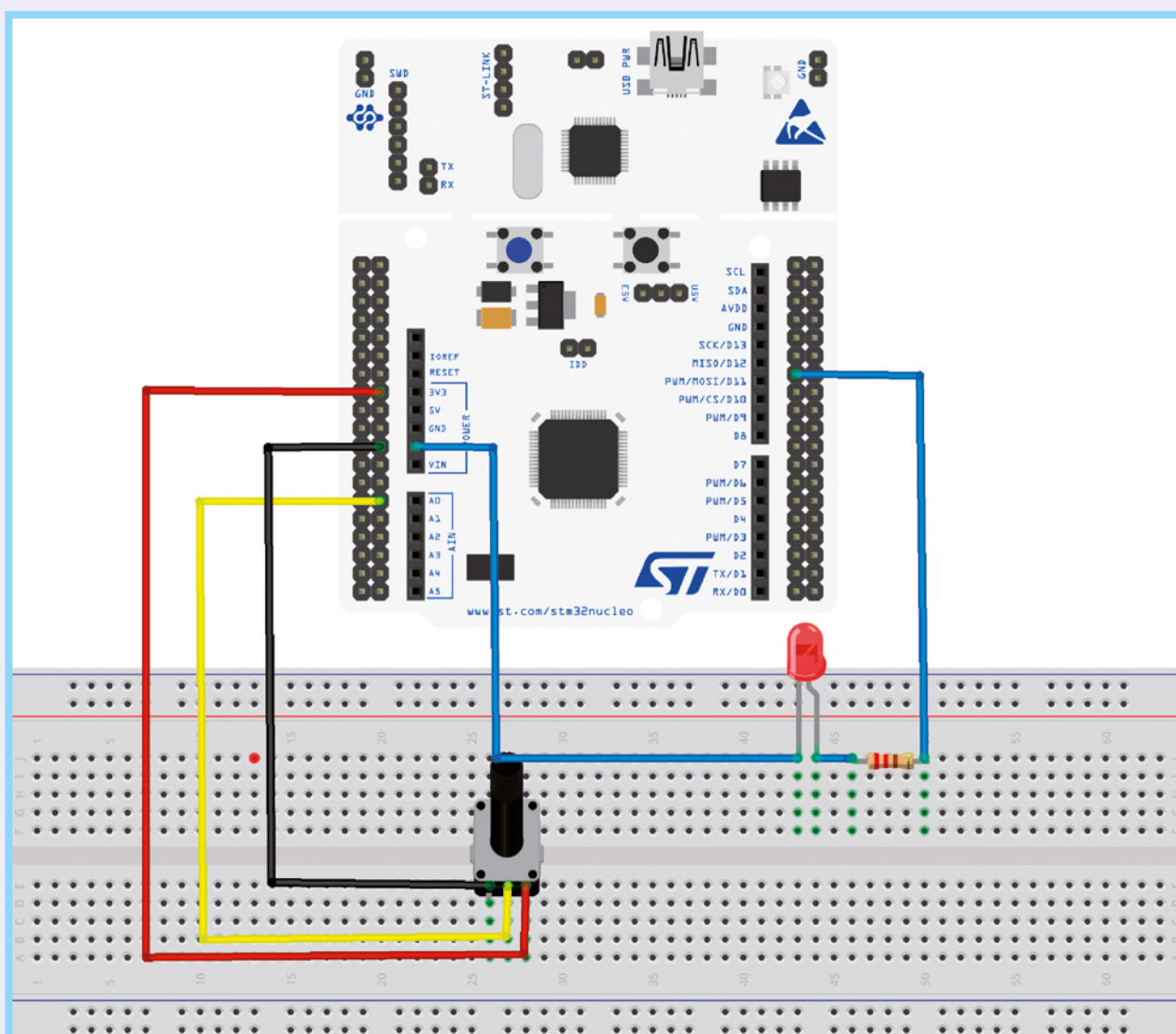


Fig. 8 Realizzazione del circuito di test

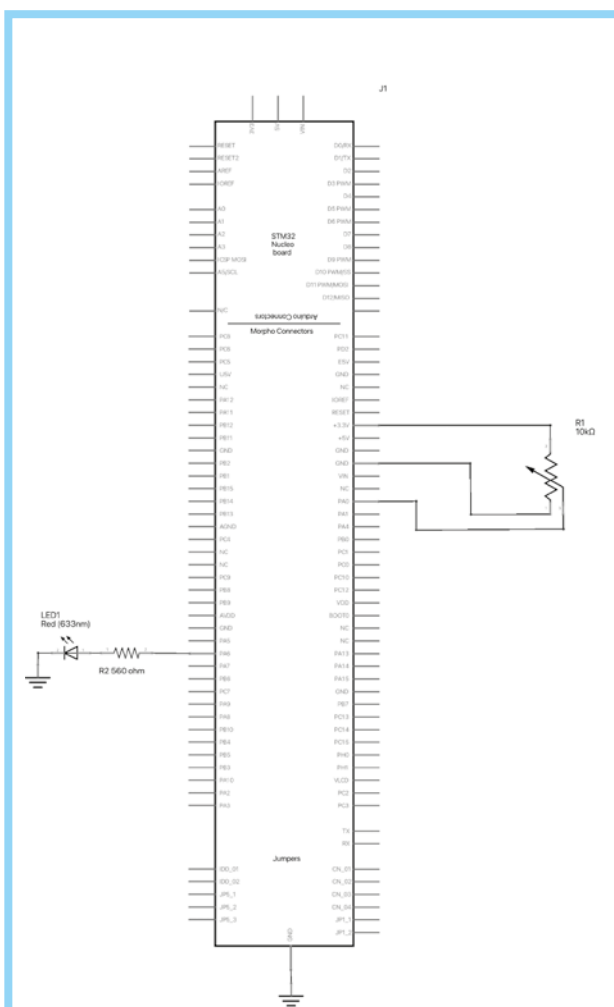


Fig. 9 Schema del circuito di test

Fig. 10 Configurazione di TIM3

che verrà memorizzato nell'apposito registro dall'istruzione di riga 73 '_HAL_TIM_SET_COMPARE' (Fig. 13).

In Fig. 14 possiamo osservare il listato di istruzioni della funzione 'main' con l'attivazione del PWM e della conversione A/D e il ciclo di controllo all'interno della 'while()' che, dalla lettura dell' ADC, calcola il duty cycle in percentuale e richiama la funzione 'dutyCycle' che aggiorna il registro Pulse ogni 100 ms.

Anche in questo programma ci siamo avvalsi dell'interrupt di gestione del convertitore A/D che segnala la fine della conversione e fornisce la corretta tempistica per la gestione

dei calcoli (Fig. 15).

Ruotando il potenziometro otterremo una variazione di tensione compresa tra 0 e 3.3V, corrispondente ad una conversione digitale da 0 a 4095.

La variabile setDutyCycle assumerà valori compresi tra 0 e 1000 che modificheranno il duty cycle da 0 a 100% con conseguente variazione dell'illuminazione del led.

Con questo esempio si è voluto dare una panoramica utile per comprendere la gestione della periferica PWM e fornire le conoscenze adatte per parametrizzare correttamente i registri associati a questa funzione.

```

/* USER CODE BEGIN PV */
_Bool end_conversion = FALSE; // Al termine della conversione viene portata a livello 1
uint16_t adcResult; // Risultato della conversione digitale
float duty; // valore calcolato del duty cycle in base alla posizione del potenziometro
/* USER CODE END PV */

```

Fig. 11 File main.c : definizione delle variabili

```

/* USER CODE BEGIN PFP */
void dutyCycle(float);
/* USER CODE END PFP */

```

Fig. 12 File main.c : definizione del prototipo della funzione

LA PERIFERICA UNIVERSAL ASYNCHRONOUS SERIAL COMMUNICATIONS:

Introduzione

Questa parte del corso è dedicata all'interfaccia più utilizzata nel mondo dei microcontrollori per la trasmissione e ricezione dati tra dispositivi esterni e la MCU: il bus seriale.

Attraverso il bus seriale due device possono attivare un flusso di segnali digitali su un singolo filo rispettando i segnali di sincronizzazione.

Le schede Nucleo sono dotate di più porte seriali USART e UART con la possibilità di avere anche i controlli hardware del flusso di dati.

MODALITÀ UART - USART

L'interfaccia seriale si presenta nelle MCU delle schede Nucleo in due distinte modalità: USART e UART (Universal Synchronous/Asynchronous Receiver/Transmitter).

La differenza tra le due modalità risiede proprio nel sistema di sincronizzazione che viene adottato.

In una trasmissione sincrona i due device, che chiameremo trasmettitore e ricevitore, utilizzano una linea di clock che permette la sincronizzazione e la velocità della trasmissione. In questo caso viene utilizzata la modalità USART.

Tra i due device vengono collegate tre linee dati: Tx, RX e clock (Figura 17).

Utilizzando la modalità UART le linee dati sono solamente due e la sincronizzazione avviene mediante segnali di START e STOP aventi una durata ben precisa.

La Figura 16 mostra la trasmissione di un byte in modalità UART inglobato tra i due segnali di START e STOP. La durata di entrambi i segnali ha un valore pari a

$$T1 = 1.5 * T2$$

dove T2 è la durata di un bit.

```

67 /* USER CODE BEGIN 0 */
68 void dutyCycle(float duty)
69 {
70     uint16_t autoReload, setDutyCycle;
71     autoReload = __HAL_TIM_GET_AUTORELOAD(&htim3); // lettura da registro auto reload
72     setDutyCycle = ((autoReload + 1) * duty / 100); // calcolo del nuovo duty cycle in base alla lettura A/D
73     __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, setDutyCycle); // configurazione registro pulse
74 }
75 /* USER CODE END 0 */

```

Fig. 13 File main.c : listato della funzione dutyCycle()

```

107 /* USER CODE BEGIN 2 */
108 HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1); // attiva il PWM
109 HAL_ADC_Start_IT(&hadc1); // attiva la routine di interrupt
110
111 /* USER CODE END 2 */
112
113 /* Infinite loop */
114 /* USER CODE BEGIN WHILE */
115 while (1)
116 {
117     /* USER CODE END WHILE */
118
119     /* USER CODE BEGIN 3 */
120     if(end_conversion == TRUE)
121     {
122         adcResult = HAL_ADC_GetValue(&hadc1); // lettura registro e memorizzazione su variabile
123         duty = adcResult * (100/RESOLUTION_BIT); // calcolo duty cycle in percentuale
124         dutyCycle(duty); // richiama funzione settaggio pwm
125         end_conversion = FALSE; // reset variabile booleana
126         HAL_Delay(100); // attesa 100 ms
127         HAL_ADC_Start_IT(&hadc1); // Reimposta le condizioni per un nuovo interrupt
128     }
129 }
130
131 }
132 /* USER CODE END 3 */
133 }

```

Fig. 14 File main.c : listato del loop while(1)

```

309 /* USER CODE BEGIN 4 */
310 void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc1)
311 {
312     end_conversion = TRUE; // Attiva la variabile essendo stata completata la conversione A/D
313 }
314
315 /* USER CODE END 4 */

```

Fig. 15 File main.c : gestione dell'interrupt del convertitore A/D

I tempismi di trasmissione vengono determinati all'apertura del canale, con la definizione della presenza di segnali di start, stop, parità, ecc.

Nel caso di una trasmissione asincrona, entrambi i dispositivi coinvolti nella trasmissione dati concordano sul tempo necessario per trasmettere un singolo bit e sulla modalità di inizio e

fine trasmissione per campionare i bit trasmessi.

Nel nostro caso analizzeremo esclusivamente la trasmissione seriale asincrona che, come visibile in **Fig. 16**, inizia con un bit di START, rappresentato dal livello logico basso.

Il fronte discendente viene rilevato dal ricevitore e trascorso il tempo T1 inizia il campionamento dei bit. Viene trasmesso

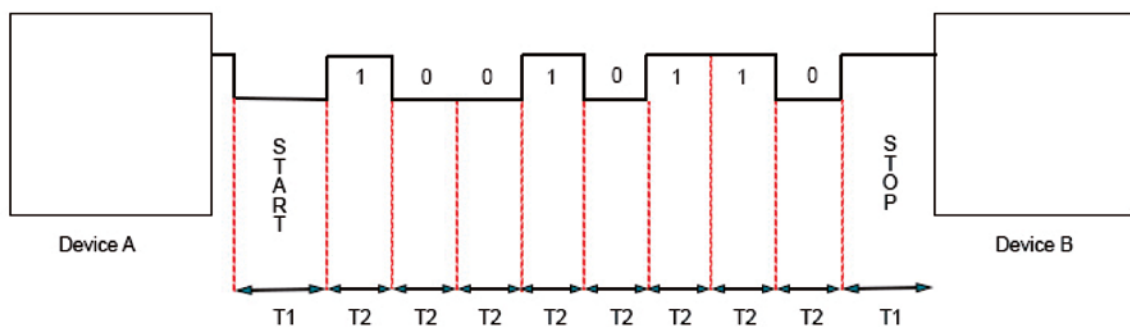


Fig. 16 Diagramma temporale di una trasmissione UART

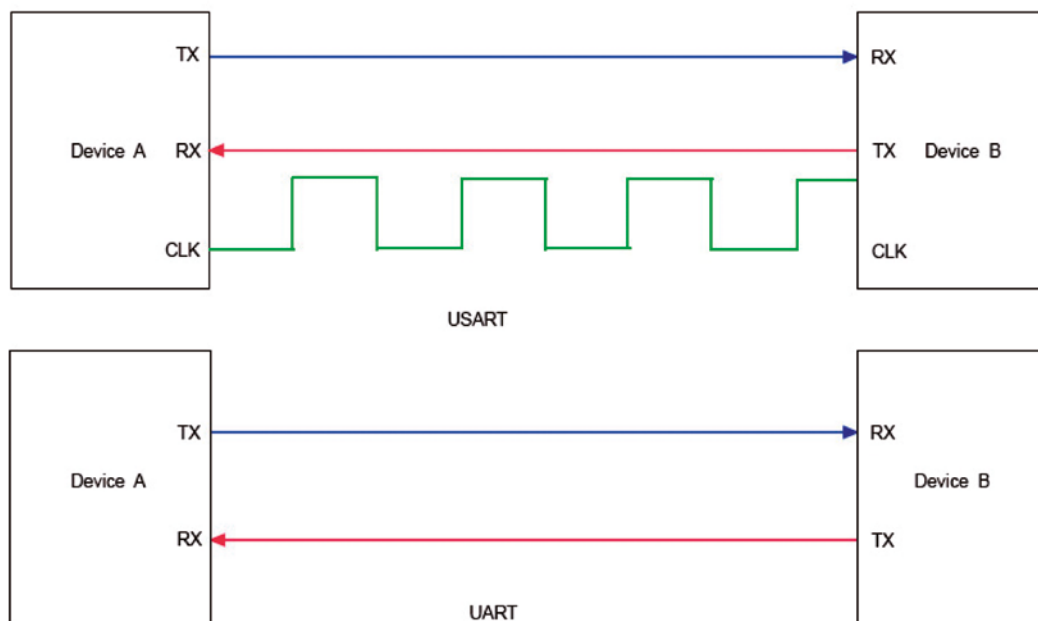


Fig. 17 Modalità di trasmissione USART e UART

NUCLEO P/N	USART + UART	USART#	HW Flow Control RS232	HW Flow Control RS485
NUCLEO-F446RE	4 + 2	USART1/2/3	SI	-
		USART6	-	-
		UART4/5	SI	-
NUCLEO-G474RE	3 + 2	USART1/2/3	SI	SI
		UART4/5	SI	SI
NUCLEO-F401RE	3 + 0	USART1/2	SI	-
		USART6	-	-
NUCLEO-L476RG	3 + 2	USART1/2/3	SI	SI
		UART4/5	SI	SI
NUCLEO-L152RE	3 + 2	USART1/2/3	SI	-
		UART4/5	-	-

Tabella 1 Schede Nucleo tipo F, L, G: numero di porte seriali presenti e controlli hardware disponibili

per primo il bit meno significativo (LSB) e vengono campionati otto bit di dati. Può essere trasmesso un bit di parità opzionale (per il controllo degli errori dei bit di dati).

Spesso questo bit viene omesso se si presume che il canale di trasmissione sia privo di rumore o se ci sono controlli degli errori più in alto nei livelli del protocollo.

La trasmissione viene terminata da un bit di STOP, di livello alto con durata T1.

In **Tabella 1** abbiamo raccolto alcuni tipi di MCU presenti su schede Nucleo tipo F, L, G, individuando per ognuna il

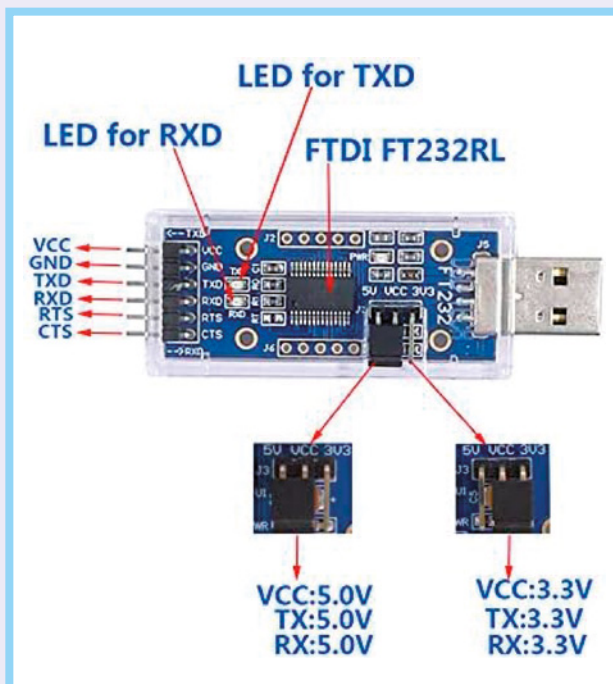


Fig. 18 Adattatore seriale UART/USB 3.3V/5V

numero di porte seriali disponibili e indicando la presenza o meno dei segnali di controllo hardware per connessione RS232 e RS485. La periferica UART di un STM32 utilizza i livelli di tensione degli I/O della MCU per trasmettere i dati verso l'esterno, per cui il livello di tensione sarà pari a V_{DD} e cioè circa 3.3 V.

Per poter comunicare con periferiche esterne che utilizzano segnali TTL a 5V come standard di comunicazione seriale, possiamo fare affidamento a interfacce seriali come la FT232RL (**Fig. 18**) che consente di collegare la porta UART di un MCU e adattarne i segnali per un'interfaccia USB. Utilizzeremo questo adattatore per interfacciare la scheda Nucleo con il computer utilizzando l'UART e la porta USB del dispositivo (**Fig. 19**).

UN PRIMO ESEMPIO DI COMUNICAZIONE

Il programma che ci accingeremo ad implementare, affron-

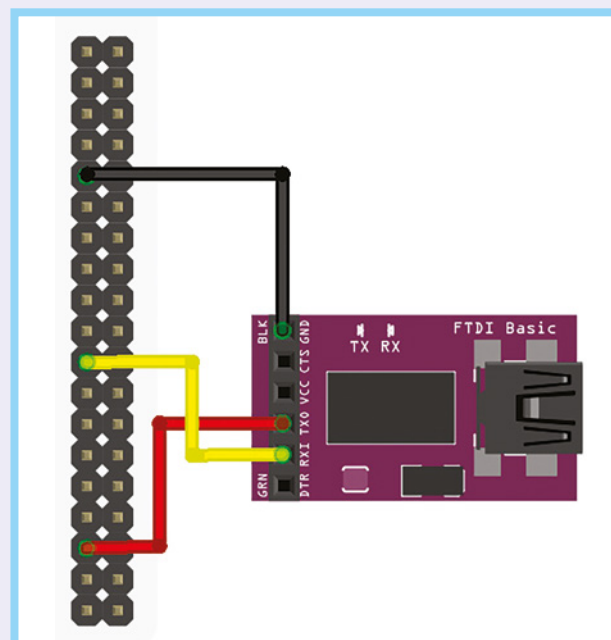


Fig. 19 Connessioni scheda Nucleo - FT232

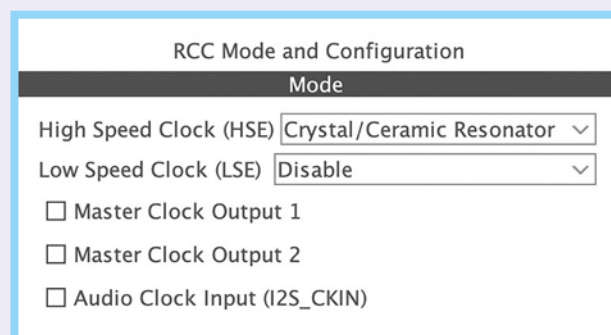


Fig. 20 Configurazione RCC

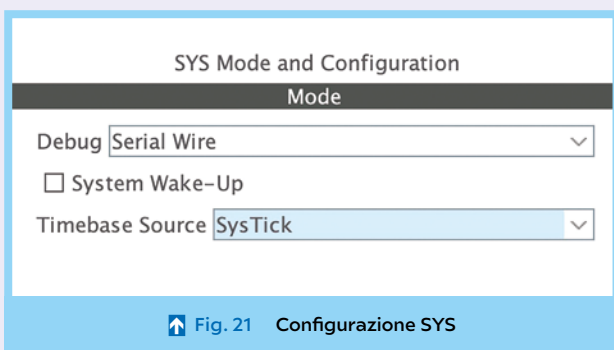


Fig. 21 Configurazione SYS

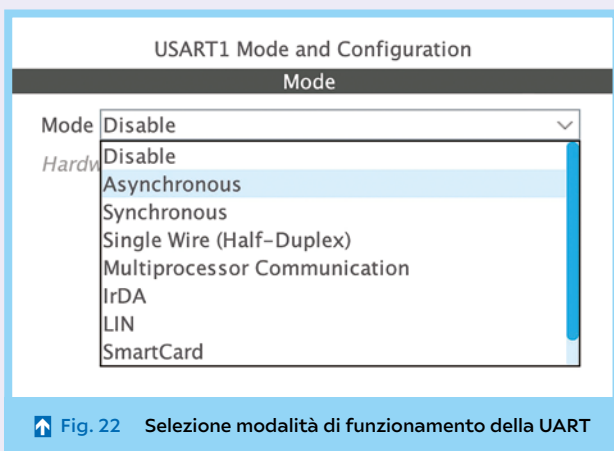


Fig. 22 Selezione modalità di funzionamento della UART

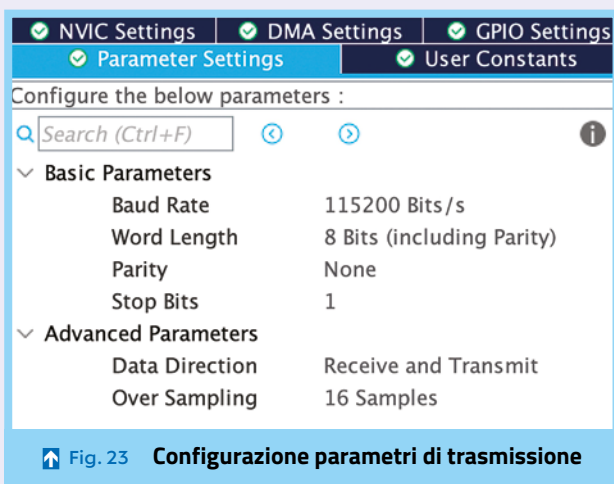


Fig. 23 Configurazione parametri di trasmissione



Fig. 24 Dettagli di configurazione delle porte PA9 e PA10

terà più step per poterci addentrare sempre più in maniera approfondita nella comprensione delle modalità di trasmissione e ricezione utilizzate nel bus seriale. Inizialmente implementeremo una semplice comunicazione tra l'interfaccia UART della scheda Nucleo e un ingresso USB del PC, utilizzando la FT232 come descritto in **Fig. 19**.

Per ottenere questo obiettivo realizziamo un nuovo progetto su STM32CubeIDE chiamandolo "F401_UART".

Le configurazioni relative alla gestione del clock e al sistema sono presentate nelle **Fig. 20, 21**, mentre per la 'Clock Configuration' possiamo fare riferimento alla **Fig. 28** della puntata 3 di questo corso.

Puntiamo invece l'attenzione sulla configurazione della porta seriale. Nel nostro caso sceglieremo la porta seriale 1 delle tre disponibili nella Nucleo F401 come indicato nella **Tabella 1**.

Per iniziare la configurazione della porta seriale, sempre nell'applicativo STM32CubeMX, apriamo il menù 'Connectivity' e selezioniamo USART1. Si aprirà la finestra 'USART1: Mode and Configuration'.

Selezionando la modalità 'Asynchronous' attiveremo la connessione di tipo UART (**Fig. 22**) seguendo le modalità di trasmissione evidenziate in **Fig. 17**.

Dopo aver selezionato la modalità asincrona, si presenterà la finestra di **Fig. 23**, dove potremo selezionare la velocità di trasmissione e gli ulteriori parametri che contraddistinguono la trasmissione seriale.

Come evidenziato in figura, sceglieremo una velocità di 115200 bits/s, 8 bits, nessuna parità, 1 bit di stop e abiliteremo la trasmissione in entrambi i sensi.

La modalità Hardware Flow Control resterà disabilitata.

Le porte interessate alla trasmissione seriale sono la PA9 e la PA10, in particolare la PA9 con funzione di trasmissione, mentre la PA10 in ricezione.

Le porte PA9 e PA10 sono entrambe presenti sul connettore CN10, rispettivamente sui pin 21 e 33. La **Fig. 24** mette in evidenza i dettagli di configurazione delle singole porte.

Ultimate le configurazioni hardware, possiamo generare il codice (menù: Project - Generate Code) e passare allo sviluppo del programma.

Inizialmente apriamo la directory 'STM32F4xx_HAL_Dri-



```

✗ HAL_UART_RegisterCallback(UART_HandleTypeDef*, HAL_UART_CallbackIDTypeDef, pUART_
✗ HAL_UART_UnRegisterCallback(UART_HandleTypeDef*, HAL_UART_CallbackIDTypeDef) : HAL
✗ HAL_UART_RegisterRxEventCallback(UART_HandleTypeDef*, pUART_RxEventCallbackTypeDe
✗ HAL_UART_UnRegisterRxEventCallback(UART_HandleTypeDef*) : HAL_StatusTypeDef
● HAL_UART_Transmit(UART_HandleTypeDef*, const uint8_t*, uint16_t, uint32_t) : HAL_StatusT
● HAL_UART_Receive(UART_HandleTypeDef*, uint8_t*, uint16_t, uint32_t) : HAL_StatusTypeDef
● HAL_UART_Transmit_IT(UART_HandleTypeDef*, const uint8_t*, uint16_t) : HAL_StatusTypeDef
● HAL_UART_Receive_IT(UART_HandleTypeDef*, uint8_t*, uint16_t) : HAL_StatusTypeDef
● HAL_UART_Transmit_DMA(UART_HandleTypeDef*, const uint8_t*, uint16_t) : HAL_StatusType

```

Fig. 25 Funzione 'HAL_UART_Transmit()' presente nel file 'stm32f4xx_hal_uart.c'

ver' e conseguentemente la directory 'Src' e selezioniamo i driver per il controllo della seriale contenuti nel file 'stm32f4xx_hal_uart.c' e osserviamo l'elenco delle funzioni nella finestra 'outline'.

Possiamo osservare che la funzione dedicata alla trasmissione, evidenziata in **Fig. 25**, è la HAL_UART_Transmit. Per individuare i driver necessari all'utilizzo di qualsiasi periferica, occorre fare riferimento, nella sezione "Project

```

225 /* USER CODE BEGIN 4 */
226 void UART_TX_NL(UART_HandleTypeDef *huart) // Trasmissione carriage return + line feed
227 {
228     HAL_UART_Transmit(huart, (uint8_t*)" \n\r", 2, HAL_MAX_DELAY);
229 }
230
231 void UART_TX_CHAR(UART_HandleTypeDef *huart, char c, int m) // Trasmissione di un carattere
232 {
233     HAL_UART_Transmit(huart, (uint8_t*)&c, 1, HAL_MAX_DELAY);
234     if(m==1)
235     {
236         UART_TX_NL(huart);
237     }
238 }
239
240 void UART_TX_INT(UART_HandleTypeDef *huart, int i, int m) // Trasmissione di una variabile intera
241 {
242     char buffer[10];
243     itoa(i, buffer, 10);
244     HAL_UART_Transmit(huart, (uint8_t*) buffer, strlen(buffer), HAL_MAX_DELAY);
245     if(m==1)
246     {
247         UART_TX_NL(huart);
248     }
249 }
250
251 void UART_TX_STR(UART_HandleTypeDef *huart, char *buffer[], int m) // Trasmissione di una stringa
252 {
253     HAL_UART_Transmit(huart, (uint8_t*) buffer, strlen(buffer), HAL_MAX_DELAY);
254     if(m==1)
255     {
256         UART_TX_NL(huart);
257     }
258 }
259
260 void UART_TX_FLOAT(UART_HandleTypeDef *huart, float p, int m) // Trasmissione di una variabile float
261 {
262     char buffer[10];
263     sprintf(buffer, "%5.2f", p);
264     HAL_UART_Transmit(huart, (uint8_t*)buffer, strlen(buffer), HAL_MAX_DELAY);
265     if(m==1)
266     {
267         UART_TX_NL(huart);
268     }
269 }
270 /* USER CODE END 4 */

```

Fig. 26 File 'main.c': listato funzioni di trasmissione


```

45 UART_HandleTypeDef huart1;
46
47 /* USER CODE BEGIN PV */
48 char tx_str[15]; // Buffer per memorizzazione stringa
49 int tx_int; // variabile intera
50 char tx_char; // variabile carattere
51 float tx_flt; // variabile float
52 int w; // variabile usata per attivare la funzione UART_TX_NL() inserire CR + LF
53 /* USER CODE END PV */
54
55 /* Private function prototypes -----*/
56 void SystemClock_Config(void);
57 static void MX_GPIO_Init(void);
58 static void MX_USART1_UART_Init(void);
59 /* USER CODE BEGIN PFP */
60
61 void UART_TX_NL(UART_HandleTypeDef *huart);
62 void UART_TX_CHAR(UART_HandleTypeDef *huart, char c, int m);
63 void UART_TX_INT(UART_HandleTypeDef *huart, int i, int m);
64 void UART_TX_STR(UART_HandleTypeDef *huart, char *buffer, int m);
65 void UART_TX_FLOAT(UART_HandleTypeDef *huart, float p, int m);
66

```

Fig. 27 File 'main.c': dichiarazione delle variabili e dei prototipi di funzione

```

93 /* Configure the system clock */
94 SystemClock_Config();
95
96 /* USER CODE BEGIN SysInit */
97
98 /* USER CODE END SysInit */
99
100 /* Initialize all configured peripherals */
101 MX_GPIO_Init();
102 MX_USART1_UART_Init();
103 /* USER CODE BEGIN 2 */
104 w = 1; // attiva la funzione UART_TX_NL() inserendo CR + LF
105 tx_flt = 1.5; // valore iniziale variabile float
106 /* USER CODE END 2 */
107
108 /* Infinite loop */
109 /* USER CODE BEGIN WHILE */
110 while (1)
111 {
112     /* USER CODE END WHILE */
113
114     /* USER CODE BEGIN 3 */
115     tx_int++;
116     UART_TX_INT(&huart1, tx_int, w);
117
118     tx_char = 't';
119     UART_TX_CHAR(&huart1, tx_char, w);
120
121     sprintf(tx_str, "Valore = %d", tx_int);
122     UART_TX_STR(&huart1, tx_str, w);
123
124     tx_flt = tx_flt + 0.2;
125     UART_TX_FLOAT(&huart1, tx_flt, w);
126
127     HAL_Delay(1000);
128 }

```

Fig. 28 File 'main.c': listato funzione main()

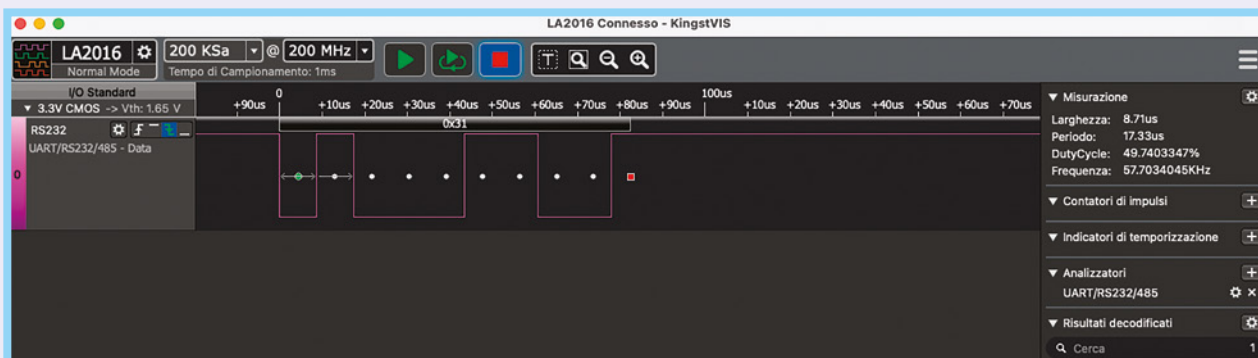


Fig. 29 Analisi trasmissione seriale

Explorer", alla directory 'Drivers' e alla sottocartella 'STM32F4xx_HAL_Driver', nel nostro caso avendo utilizzato la scheda Nucleo F401.

All'interno della cartella troviamo le sub directory 'Src' e 'Inc'; aprendo la 'Src' troviamo all'interno tutti i file dedicati alle periferiche presenti nella MCU in uso con descrizioni dettagliate sul loro utilizzo e svariati commenti.

Nel caso di eventuali approfondimenti è possibile effettuare ulteriori ricerche sul sito di STM e nella Community di STMicroelectronics.

Ritornando alla funzione di trasmissione, i parametri interessanti riguardano la struttura 'UART_HandleTypeDef', un puntatore al buffer dei dati da trasmettere, la dimensione del buffer e il tempo massimo di trasmissione, superato il quale verrà generato un errore.

In Fig. 26 presentiamo una serie di funzioni adatte alla trasmissione di variabili tra cui interi, caratteri, stringhe e float. Queste funzioni saranno richiamate durante il test e verificheremo la corretta funzionalità della trasmissione seriale. Completiamo il programma inserendo le istruzioni rappresentate nelle Fig. 26, 27 e 28 dove troviamo le dichiarazioni dei prototipi delle funzioni (righe 61-65 Fig. 27) e la dichiarazione delle variabili (righe 48-52 Fig. 27).

In Fig. 26 troviamo le funzioni dedicate a convertire i dati in formato trasmissibile, vale a dire con il proprio codice carattere ASCII.

La Fig. 28 rappresenta la routine che ogni secondo implementa le variabili e le trasmette alle funzioni dedicate.

ANALISI HARDWARE E SOFTWARE

Terminata la configurazione software, colleghiamo attraverso la ST-Link la scheda Nucleo al computer mediante cavo USB.

Premettendo che sia anche stato effettuato il collegamento UART-FT232-computer, descritto in Fig. 19, selezioniamo un programma di gestione RS232 come CoolTerm, SerialTools, PuTTY, ZeroTerm, WCHSerialPort o qualsiasi altro terminale software a disposizione e procediamo con la configurazione della porta seriale del computer che riceverà i dati dalla periferica UART.

In Fig. 29 è rappresentata la registrazione, eseguita mediante analizzatore di stati logici collegato alla porta PA9 della scheda Nucleo, della trasmissione del numero '1' attraverso la funzione UART_TX_INT().

In notazione esadecimale il carattere '1' ha il codice ASCII 0x31, che tradotto in binario è rappresentato con l'ottetto '00110001'.

Analizzando la traccia troviamo il bit di Start della durata di circa 8µs utilizzato per la sincronizzazione della trasmissione, a seguire il bit meno significativo (LSB in questo caso di valore '1'), fino al più significativo, di valore '0'. Al termine viene trasmesso il bit di Stop con segnale a livello '1'.

In Fig. 30 osserviamo le singole trasmissioni ricevute dal terminale su PC separate da un 'carriage return + line feed' generato dalla funzione UART_TX_NL().

Per ottenere un emulatore di terminale integrato nell'IDE, utile per i nostri test sulle trasmissioni seriali, proponiamo l'installazione di un emulatore di terminale disponibile direttamente nel Marketplace di Eclipse.

Per ottenerlo selezioniamo direttamente dalla barra dei

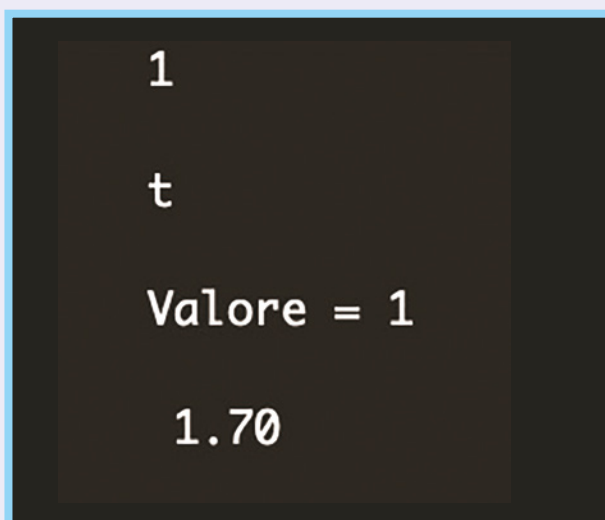


Fig. 30 Visualizzazione dei dati seriali trasmessi

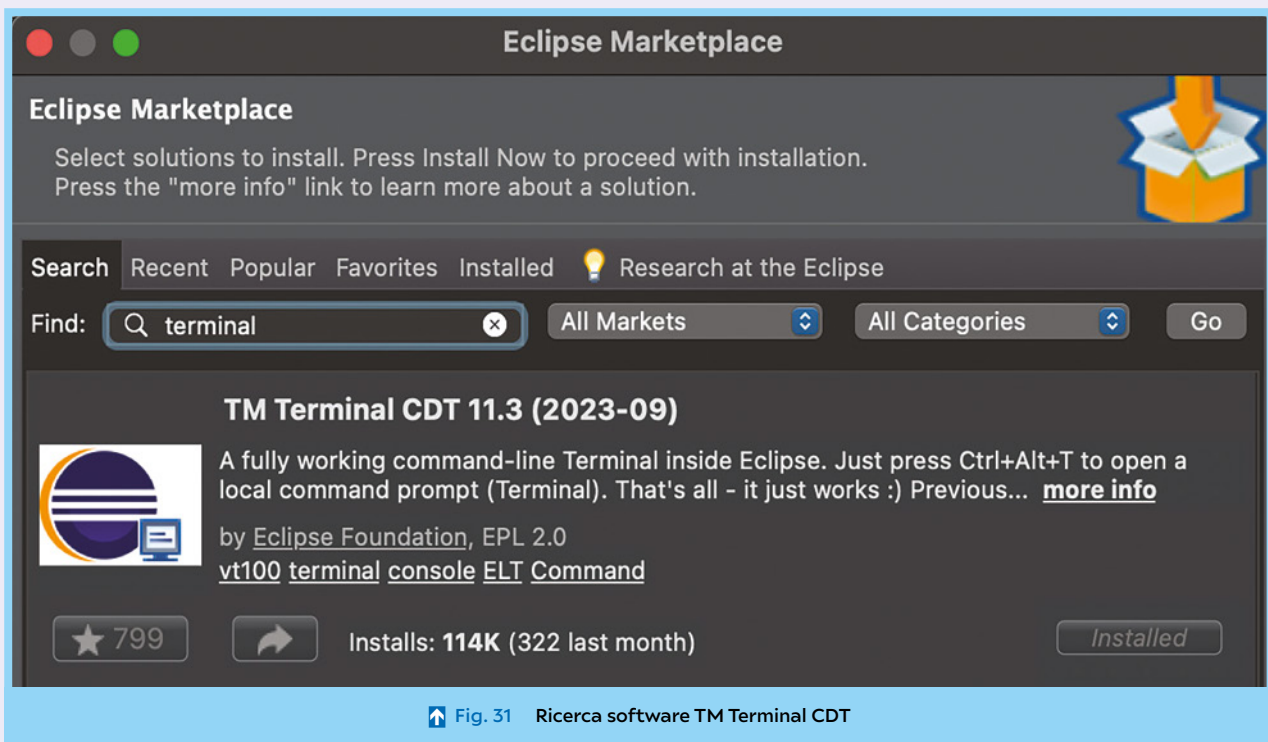


Fig. 31 Ricerca software TM Terminal CDT

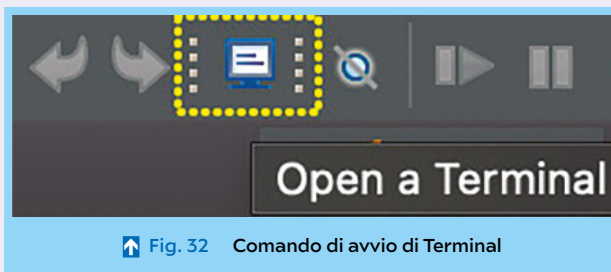


Fig. 32 Comando di avvio di Terminal

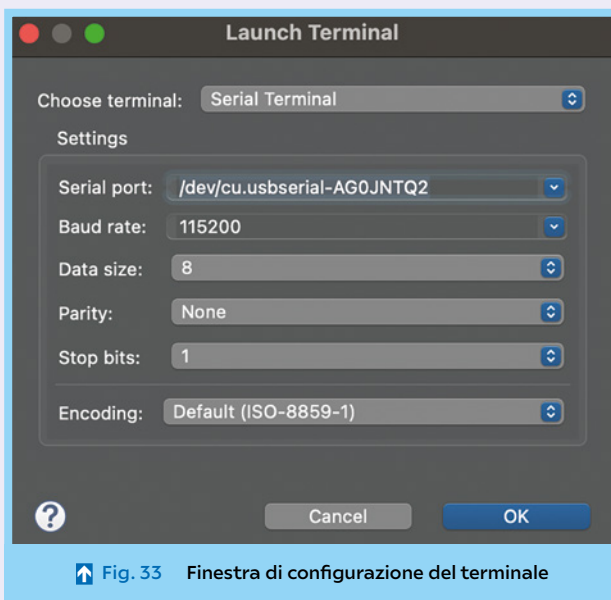


Fig. 33 Finestra di configurazione del terminale

menù " Help -> Eclipse Marketplace ".

Comparirà una finestra come in **Fig. 31** dove andremo a digitare nel text box 'Find' la parola "terminal".

Al termine della ricerca comparirà la voce "TM Terminal CDT ..." e selezioniamo il tasto "Install".

Il plugin verrà installato automaticamente e al termine verrà richiesto di riavviare l'IDE.

Come si osserva in **Fig. 32** comparirà l'icona contrassegnata con un rettangolo giallo e cliccandoci sopra si avvierà il programma dell'emulatore di terminale presentando la finestra di **Fig. 33** dove andremo a selezionare inizialmente il tipo di terminale, nel nostro caso "Serial Terminal", la porta seriale e i parametri relativi alla connessione.

Nell'IDE comparirà una nuova tab "Terminal" nella finestra di dialogo in basso dove si potranno osservare le comunicazioni seriali da e verso la scheda Nucleo.

CONCLUSIONI

In questa puntata abbiamo affrontato alcune tecniche di utilizzo della periferica PWM e abbiamo introdotto la periferica UART predisponendo un semplice test utile a familiarizzare con le finestre di configurazione e i driver di trasmissione dati. Nella prossima puntata analizzeremo due metodi di ricezione e trasmissione dati seriali sfruttando gli interrupt e il DMA.

Approfondiremo in questo modo le tecnologie messe a disposizione nelle MCU STM32 e osserveremo in dettaglio le caratteristiche prestazionali che esse permettono di raggiungere.

Impara facilmente l'elettronica!

EDUKIT 01

11

**FACILI PROGETTI
PER IMPARARE
L'ELETTRONICA
DI BASE**

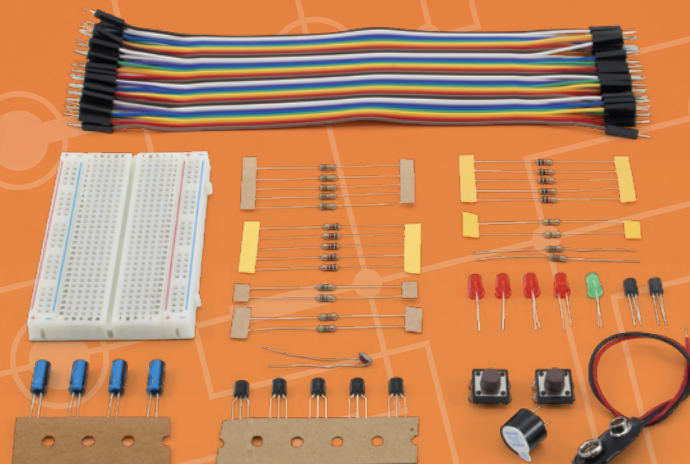
Kit di 11 esperimenti di elettronica base

Divertente, sicuro ed educativo, non richiede alcuna saldatura. Il kit contiene una breadboard 400 contatti con base autoadesiva e una serie di componenti per realizzare 11 progetti di elettronica di base. Tra i progetti: LED lampeggianti, sensore di luce, tester di polarità, circuito di start-stop, timer e molti altri.



€ 19,90

cod. EDUKIT01



Kit di 10 esperimenti con energia solare

Avvicinati facilmente al mondo dell'energia solare apprendendo e mettendo in pratica semplici circuiti. Per la realizzazione dei 10 progetti non serve alcuna attrezzatura, nemmeno il saldatore, in quanto il sistema a breadboard permette ogni genere di collegamento elettrico tra i componenti forniti, mediante l'innesco degli stessi nella piastra; basta seguire le immagini presenti nel manuale in dotazione corredato di schemi elettrici e di descrizione tecnica.

EDUKIT 02



10

**ESPERIMENTI
CON DIFFERENTI
GRADI DI
DIFFICOLTÀ**

€ 17,90

cod. EDUKIT02



Prezzi IVA inclusa.

L'inseguitore solare con Deep Learning

I sistemi fotovoltaici dotati di inseguitori solari guadagnano un 20÷30% di rendimento rispetto ai sistemi fissi montati a terra. Oltre a ciò, la progettazione e l'allineamento del layout possono tenere conto di altri criteri, come ad esempio il fabbisogno di luce di alcune varietà di piante sotto impianti agrioltaici e fotovoltaici per la biodiversità o anche l'immissione in rete in determinate ore del giorno. Nel progetto di ricerca "DeepTrack", la Zimmermann PV-Tracker GmbH, parte del gruppo Zimmermann PV-Steel, e l'Istituto Fraunhofer per i sistemi ad energia solare ISE, stanno migliorando gli algoritmi di tracciamento con un gemello digitale che utilizza il Deep Learning per calcolare approcci di controllo ottimizzati. Tra le altre cose, il gemello digitale (Digital Twin) impara dai dati del suo "vero" gemello, un inseguitore fotovoltaico costruito da Zimmermann PV-Tracker, che si trova presso l'Outdoor Performance Lab del Fraunhofer ISE a Merdingen vicino a Friburgo. Secondo la Roadmap tecnologica internazionale per il fotovoltaico pubblicata dalla Federazione tedesca degli ingegneri (VDMA), in futuro il 60% di tutte le centrali fotovoltaiche nel mondo disporrà di un inseguitore. Nei paesi con alti livelli di radiazione solare, come la Spagna, tali sistemi rappresentano già la maggior parte degli impianti fotovoltaici a terra di nuova costruzione. Nell'ambito del progetto DeepTrack, un sistema fotovoltaico con inseguimento è stato posato sul campo di prova esterno del Fraunhofer ISE per ottenere misurazioni in condizioni reali. Sulla base dei risultati delle misurazioni è stato creato un gemello digitale che abbina gli strumenti di monitoraggio e modellazione del fotovoltaico alle previsioni meteorologiche grazie al deep learning. Le posizioni di tracciamento ottimali dei moduli sono mappate per diverse situazioni.

www.fraunhofer.de



Sfere sensorizzate aumentano la portata della rete elettrica

Le tradizionali linee di trasmissione ad alta tensione rappresentano oggi un collo di bottiglia nella rete elettrica, ma cosa accadrebbe se una semplice modifica potesse aumentare la loro capacità fino al 40%? Questo è ciò che sta esaminando la Great River Energy, un'azienda energetica (o Utility, come usa dire oggi...) del Minnesota (USA). Dopo un progetto pilota di successo lo scorso anno, l'azienda sta installando 52 "neuroni" con sensori intelligenti, A.K.A. sulle proprie linee di trasmissione ad alta tensione. Queste sfere sono realizzate dalla società Heimdall Power con sede a Oslo. Ogni neurone contiene sensori che aiutano gli opera-

tori a gestire la trasmissione di potenza in modo più efficiente.

Lo scopo del progetto è affrontare un problema che affligge le linee di trasmissione: perdono capacità man mano che diventano più calde. All'aumentare della temperatura dell'aria, gli operatori riducono la quantità di elettricità inviata attraverso le linee per ridurre al minimo le perdite ed evitare possibili interruzioni di corrente dovute al surriscaldamento. Anche le linee ad alta tensione hanno un problema di "cadute"; i fili si espandono e si allungano man mano che diventano più caldi. Ciò avvicina la parte inferiore degli anelli tra le torri al suolo, dove potrebbero correre il rischio di entrare in contatto con alberi o edifici.

La pratica standard è stata quella di fissare limiti in base alla stagione.



Questi limiti sono spesso basati su condizioni prudenti, basate sullo scenario peggiore, tuttavia tale approccio lascia inutilizzata la potenziale capacità di trasmissione. I neuroni che la Heimdall ha battezzato "sfere magiche" hanno le dimensioni di un pallone da calcio e sono montati su un cavo della linea di trasmissione. Possono anche essere installati da un drone autonomo in meno di due minuti.

I sensori monitorano la temperatura e la corrente elettrica. Possono anche svolgere altri compiti importanti sulla linea, come il rilevamento dei guasti e il monitoraggio del ghiaccio. I dati raccolti dai neuroni vengono quindi elaborati in Cloud e consegnati agli operatori in modo che possano utilizzare l'intera capacità delle linee di trasmissione sulla base di dati in tempo reale anziché di ipotesi migliori eseguite a distanza, il che spesso, per motivi di sicurezza, devono essere stime prudenti. Tali deficit di informazione, afferma l'azienda, possono lasciare le linee elettriche funzionanti a velocità inferiori a quelle di picco e la capacità di trasmissione del mondo reale regolarmente inutilizzata.

Storicamente, le tecnologie per monitorare le linee di trasmissione sono state costose e difficili da implementare, e quindi utilizzate solo su scala limitata su poche linee selezionate. Gli operatori di rete hanno altre opzioni, ma sono più costose e possono richiedere molto più tempo per essere implementate. Costruire una nuova linea ad alta tensione in parallelo per aumentare la capacità può costare tra 1,17 milioni di dollari e 8,62 milioni di

dollari per miglio, secondo l'Energy Information Administration degli Stati Uniti, e richiedere anni per la costruzione. Un altro approccio è quello di "ricondurre" le linee ad alta trasmissione esistenti sostituendo i cavi esistenti con "conduttori aerei avanzati" che possono trasportare fino a 10 volte l'elettricità dei cavi standard.

Secondo il Dipartimento dell'Energia degli Stati Uniti, un progetto del genere può costare la metà o meno di una nuova linea di trasmissione e richiedere una frazione del tempo per essere realizzato. Al contrario, i neuroni di potenza di Heimdall, afferma Festervoll, possono essere utilizzati rapidamente e a costi relativamente bassi. Il risultato, afferma, è un aumento significativo della capacità delle linee di trasmissione a una frazione del costo e in un tempo molto più breve rispetto alla costruzione di nuove linee di trasmissione o alla sostituzione dei cavi esistenti.

Great River Energy ha iniziato il suo viaggio con un progetto pilota con quattro Heimdall Power Neurons, afferma Festervoll. Utilizzando valutazioni dinamiche delle linee per regolare la trasmissione di potenza in linee selezionate, l'azienda è riuscita ad aumentare la capacità di trasmissione del 42,8%. Costruire ulteriori linee di trasmissione per ottenere un aumento equivalente richiederebbe anni e sarebbe molto più costoso. Secondo un rappresentante dell'azienda, l'impiego delle sfere sensoriali costa solo dal 2 al 5% del costo di costruzione di una nuova linea.

greatriverenergy.com

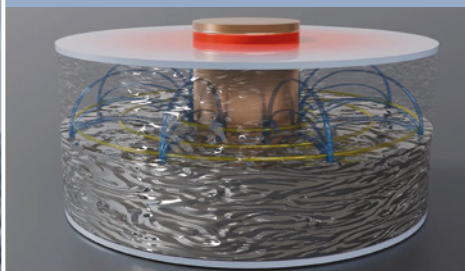


L'effetto termoelettrico anche nei liquidi

I dispositivi termoelettrici convertono l'energia termica in energia elettrica e viceversa, tuttavia per 200 anni, tanti ne sono trascorsi da quando il fenomeno è stato osservato, non era mai stato riscontrato in un sistema interamente liquido. Oggi, però un gruppo di ricercatori dell'École Normale Supérieure (ENS) di Parigi ha osservato la termoelettricità all'interfaccia tra due metalli liquidi; questa scoperta potrebbe essere utilizzata per creare nuovi dispositivi per recuperare energia dal calore di scarto e aiutare a migliorare la progettazione delle batterie a metallo liquido.

I ricercatori hanno studiato due metalli liquidi a temperatura ambiente: gallio e mercurio. Gli esperimenti sono stati condotti in un cilindro con pareti refrigerate, al cui centro è stato posizionato un riscaldatore cilindrico più piccolo. I ricercatori hanno versato del mercurio liquido denso nel cilindro esterno, quindi lo hanno ricoperto con uno strato di gallio liquido più leggero. Hanno riscaldato i liquidi dall'interno e raffreddato le pareti esterne del cilindro, creando un gradiente di temperatura lungo l'interfaccia tra i due metalli. Fili immersi nei metalli liquidi misuravano i campi elettrici risultanti. I ricercatori hanno rilevato una corrente elettrica in risposta alla differenza di temperatura, che risaliva attraverso il mercurio dal lato caldo della camera a quello freddo, quindi attraversava l'interfaccia nel gallio. Da lì, la corrente scorreva dal lato freddo del gallio a quello caldo, poi di nuovo nel mercurio e così via; vi erano però punti dove non c'era corrente, cosa che non accade nei solidi. I ricercatori hanno anche esposto l'insieme a un campo magnetico, allorché il liquido ha iniziato a ruotare nel serbatoio; il movimento può ridurre le reazioni collaterali nella batteria, consentendo una ricarica più rapida e una potenza maggiore.

www.ens.psl.eu/en



L'UAV decolla e atterra in verticale

Airbus ha arricchito la propria offerta di soluzioni UAS con Flexrotor, l'UAV (velivolo senza pilota) con funzionalità VTOL (Vertical Take Off and Landing) e capace di trasportare una massa di 8 kg. Progettato per l'intelligence, la sorveglianza, l'acquisizione di bersagli e la ricognizione (ISTAR) questo UAV si distingue per la capacità di decollo e atterraggio verticale da un'area di lancio che misura solo 3,7 x 3,7 m; una volta sollevatosi a sufficienza, il rotore inclinabile passa dalla modalità verticale a quella orizzontale, con un'apertura alare di 3 metri. Flexrotor può essere impiegato sul campo, lanciato da ponti ristretti o addirittura utilizzato a bordo di piccole motovedette, garantendo un'autonomia di volo da 14 a 30 ore. L'UAV trova impiego in vari ambiti: dalla ricognizione militare a



quella in aree dove un velivolo con piloti non potrebbe volare senza correre troppi rischi, al telerilevamento (ad esempio equipaggiato con un rilevatore di metano (per monitorare lo scioglimento dei ghiacci polari) fino alla realizzazione di un ponte radio volante.

www.airbus.com

Rilevare la glicemia senza pungere

Per le persone con diabete, i misuratori di glucosio sono uno strumento prezioso, tuttavia l'attuale generazione di questi biosensori rileva i livelli di glucosio con sottili filamenti metallici inseriti nel tessuto sottocutaneo, lo strato più profondo della pelle dove si trova la maggior parte del grasso corporeo.

La Biolinq sta ora sviluppando un nuovo tipo di sensore del glucosio che sfrutta l'attività metabolica negli strati più superficiali della pelle, utilizzando una serie di microsensori elettrochimici per misurare il glucosio e altre sostanze chimiche nel corpo appena sotto la superficie della pelle. Il sensore Biolinq è il primo sensore intradermico al mondo completamente autonomo: mentre altri monitor del glucosio richiedono uno smartphone o un altro lettore per raccogliere e visualizzare i dati, Biolinq include un display a LED per mostrare quando il glucosio rientra nella normalità (indicato da una luce blu) o è patologico (luce gialla).

Il dispositivo di Biolinq è costituito da una serie di microaghi incisi su un wafer



di silicio utilizzando la produzione di semiconduttori (i filamenti di altri sensori di glucosio vengono inseriti con un ago introduttore). Ogni chip ha un ingombro di 2 x 2 millimetri e contiene sette microaghi indipendenti, che sono rivestiti con membrane attraverso un processo

simile alla galvanica nella creazione di gioielli. Una sfida che l'industria ha dovuto affrontare è garantire che i microsensori non si rompano su scala così ridotta.

La miniaturizzazione consente di percepire più vicino alla superficie della

pelle, dove è presente un elevato livello di attività metabolica. Grazie a questa versatilità, combinata con l'uso di una serie di sensori, il dispositivo in fase di sviluppo può anche monitorare il lattato, un importante indicatore dell'affaticamento muscolare. Con l'aggiunta di un terzo punto dati, i chetoni (prodotti quando il corpo brucia i grassi), Biolinq mira ora a realizzare un pannello metabolico su un unico chip semiconduttore.

www.biolinq.com

Nvidia spinge sull'AI per ADAS e robotica

Nvidia ha creato Omniverse Cloud Sensor RTX, un set di microservizi che consentono la simulazione di sensori fisicamente accurata per accelerare lo sviluppo di macchine completamente autonome di ogni tipo. I sensori forniscono a veicoli autonomi, umanoidi, manipolatori industriali, robot mobili e spazi intelligenti i dati necessari per comprendere il mondo fisico e prendere decisioni informate. Con Omniverse Cloud Sensor RTX, gli sviluppatori possono testare la percezione dei sensori e il software AI associato su larga scala in ambienti virtuali realistici e fisicamente accurati prima dell'implementazione nel mondo reale, migliorando la sicurezza e risparmiando tempo e costi.

Costruito sul framework OpenUSD e basato sulle tecnologie di ray-tracing e rendering neurale NVIDIA RTX, Omniverse Cloud Sensor RTX accelera la creazione di ambienti simulati combinando dati del mondo reale provenienti da video, telecamere, radar e lidar con dati sintetici. Anche per scenari con dati reali limitati, i microservizi possono essere utilizzati per simulare un'ampia gamma di attività, ad esempio se un braccio robotico funziona correttamente, il nastro trasportatore dei bagagli di un aeroporto funziona, il ramo di un albero blocca una carreggiata, il nastro trasportatore di una fabbrica è in movimento oppure nelle vicinanze si trova un robot o una persona.

Omniverse Cloud Sensor RTX consentirà inoltre ai produttori di sensori di convalidare e integrare i gemelli digitali dei loro sensori in ambienti virtuali.

www.nvidia.com



SERVIZIO DI PROGETTAZIONE

L'area R&D di Futura Group studia, progetta e realizza prototipi per differenti settori utilizzando tecnologie innovative. Il team Futura Group è composto, oltre che dal personale dei nostri laboratori tecnici, anche da un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore.



SERVIZIO DI REALIZZAZIONE PCB

Grazie al Servizio di realizzazione PCB potrai ricevere i tuoi circuiti stampati semplicemente caricando sul nostro sito il tuo Gerber e scegliendo le dimensioni e le caratteristiche preferite. È possibile ordinare schede a una singola faccia oppure multistrato, scegliendo anche il colore del solder.



Trasformiamo le tue idee in realtà!

FUTURA ELETTRONICA È IL TUO PARTNER DI RIFERIMENTO PER MOLTEPLICI SERVIZI. I NOSTRI TECNICI SONO A TUA DISPOSIZIONE PER FORNIRE TUTTE LE INFORMAZIONI E UNA VALUTAZIONE PERSONALIZZATA IN BASE ALLE TUE NECESSITÀ.

SERVIZIO TAGLIO LASER

Futura Elettronica offre la possibilità di realizzare mediante taglio laser i tuoi progetti in plexiglass (o metacrilato) con spessore massimo di 10 mm e con dimensioni massime lavorabili di 130x90 cm. Carica sul nostro sito il tuo file .DXF, scegli le dimensioni e le caratteristiche preferite.



SERVIZIO STAMPA 3D

Futura Elettronica mette a disposizione le proprie stampanti 3D FDM (fused deposition modeling), per chi intende realizzare un oggetto tridimensionale con dimensioni fino a 40x40x40 cm partendo da un modello 3D digitale. Carica sul nostro sito il tuo file .STL, scegli le dimensioni e le caratteristiche preferite.



**FUTURA
ELETTRONICA®**

www.futurashop.it

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

Coppia di ricetrasmittitori 8 canali PMR446

Comunica a costo zero con i nuovi ricetrasmittitori PMR 446.

Ideali per: tempo libero, escursione, lavoro, arrampicata, caccia, softair,
o come interfono all'interno dell'abitazione.

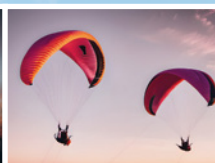
PORTATA
FINO A **5Km**
IN SPAZI APERTI



Coppia di ricetrasmittitori a 8 canali operanti sulla banda UHF. Appartengono alla categoria PMR446 e non necessitano di licenza d'uso. Dispongono di display retroilluminato, 10 tipi di suonerie di chiamata, VOX per comunicare a mani libere, funzione SCAN, tasto CALL di chiamata e mini torcia a led.



Escursionismo



Sport



Lavoro



Interfono

€ 35,⁵⁰

cod. WTA446

Prezzo IVA inclusa.



IL SALOTTINO DEL LIBRO TECNICO

Ultimi libri presentati

SCARICABILE
GRATUITAMENTE



CORSO DIGITALE

cod. RP2040



€ 16,50

cod. S5730



€ 9,90

cod. S5727

Prezzi IVA inclusa.

Scopri tutti su futurashop.it



€ 10,90

cod. S5723



€ 14,90

cod. S5726



€ 13,90

cod. S5725

Cerca, trova, proteggi! Localizzatori satellitari per ogni necessità!

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

LOCALIZZATORE 2G GPS/GSM/GPRS



**IL PIÙ
VENDUTO**

cod. FR864
€48,00

Dispone di base magnetica, ascolto ambientale e batteria da 1500mA. Servizio di localizzazione tramite App/ Piattaforma-Web/SMS e App dedicata alla configurazione. Grado di protezione IP65. Autonomia 3-5 giorni circa, in modalità tracking ad intervalli di un minuto.

LOCALIZZATORE GPS 4G LTE-2G - BATTERIA 3000 mA

**4G
LTE**



cod. FR851
€96,00



Prezzi IVA inclusa.

LOCALIZZATORE GPS 4G LTE-2G BATTERIA 1500 mA



Localizzatore GPS 4G di dimensioni compatte con grado di protezione IP65 che consente il monitoraggio in tempo reale di veicoli e persone tramite browser web o APP gratuita. Autonomia 2-4 giorni circa, in modalità tracking ad intervalli di un minuto, 5-7 giorni power saving mode, 10-15 giorni in modalità ibernazione.

**4G
LTE**

cod. FR865
€74,00

Ideale per l'uso in auto e personale (ad esempio per escursionisti), dotato di tracciamento in real-time, ascolto ambientale, tasto SOS, Geofence. Grado di protezione IP65. App dedicata per il tracciamento e la configurazione. Autonomia 5-7 giorni circa, in modalità tracking ad intervalli di un minuto, 15 giorni circa in modalità ibernazione. "Hibernation mode"

LOCALIZZATORE VEICOLARE GPS 4G LTE - 2G GSM/GPRS/EDGE



**4G
LTE**

cod. FR866
€79,00

LOCALIZZATORE GPS/GSM/Wi-Fi CON FUNZIONE SOS - 4G



Dispositivo di localizzazione satellitare GPS/GSM e Wi-Fi ad alta precisione. Facile e discreto nell'installazione sul veicolo, permette di localizzarlo con precisione in tempo reale. Viene fornito con una SIM a copertura mondiale attivabile gratuitamente, con 12 mesi di servizio rinnovabile alla scadenza.

cod. TRAKIMO4G
€159,00

Si installa in auto e sfrutta la connessione internet per monitorare la posizione del veicolo mediante App in cloud. Dotato di ascolto ambientale e batteria integrata, per funzionare anche in assenza della batteria del veicolo. È dotato di una batteria agli ioni di litio da 180 mAh che offre un'autonomia necessaria da effettuare il ciclo di avvisi di allarme in assenza della tensione del veicolo.

**Informazioni dettagliate su
www.futurashop.it**