

Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXVII - n. 258 - OTTOBRE 2021



person: 1.0



Allarme anti-intrusione con AI

■ RFID con RC522

■ Generatore 100Hz

■ Sonde differenziali per alta tensione

■ Micro oscilloscopio

■ Conversione A/D con Fishino32

■ Totem RoboBoard X4

■ Smart Mirror con Alexa

Lettura Green Pass con OpenCV e Python



Piatto per saldatura a rifusione



CONOSCERE
E USARE



Node-RED

ROME

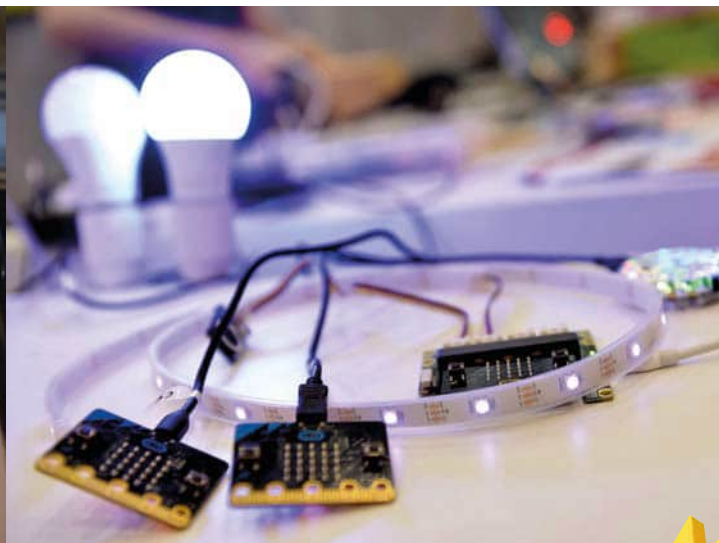
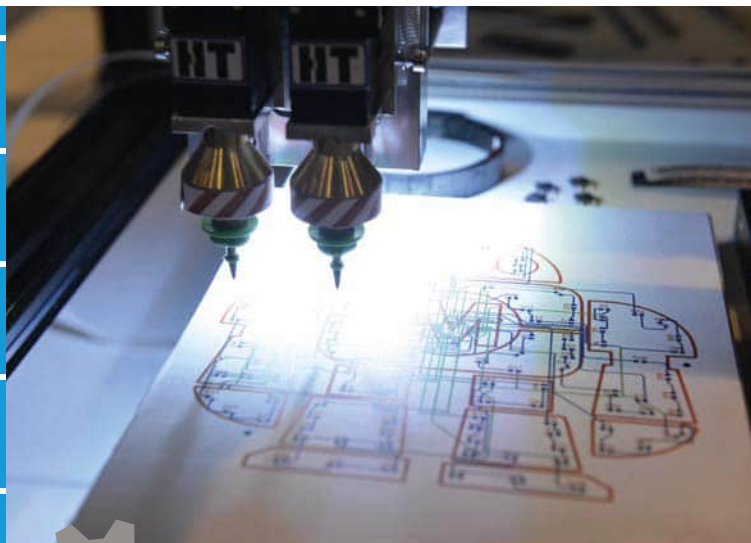
Maker Faire®

THE EUROPEAN EDITION

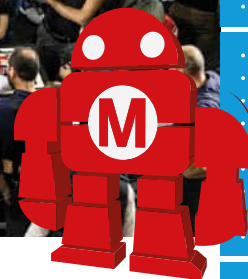
9^a edizione

8 - 10 OTTOBRE 2021

MAKERFAIREROME.EU



IL PIÙ GRANDE EVENTO EUROPEO SULL'INNOVAZIONE



MAKERFAIREROME.EU

#MFR2021  **Maker Faire Rome**

CON IL SUPPORTO DI



IN PARTNERSHIP

INAIL

IN COLLABORAZIONE CON



MAIN PARTNER



GOLD PARTNER



MEDIA PARTNER



PROMOSSO
E ORGANIZZATO DA





Internet of Value

Sulle nostre pagine abbiamo presentato molti progetti basati sull'IoT (Internet of Things) ossia l'Internet delle cose, e non c'è mancata l'occasione di affrontare l'argomento Blockchain. Per Internet delle cose, si intende la tecnologia che consente a determinati oggetti di interagire tra loro per tramite di un'automazione e scambiandosi dei dati. L'obiettivo dell'IoT è quello di migliorare le prestazioni di prodotti, creando una rete di infrastrutture in grado di elaborare, registrare e processare dati, restituendo all'utente un servizio personalizzato. La tecnologia blockchain, semplificando, è paragonabile ad un registro pubblico che in automatico, e con modalità sicure e trasparenti, è in grado di registrare un rilevante numero di

transazioni in digitale.

L'integrazione tra IoT e Blockchain ha dato vita all'IoV (Internet of Value) consentendo di valorizzare i dati, rendendoli tracciabili, trasparenti e verificabili da tutti.

Sebbene il mondo finanziario sia stato il primo ad aver utilizzato tale tecnologia, le potenzialità della blockchain iniziano ad essere apprezzate anche

in altri settori. In particolare, l'integrazione tra IoT e Blockchain ha dato vita all'IoV (Internet of Value) consentendo di valorizzare i dati, rendendoli tracciabili, trasparenti e verificabili da tutti gli oggetti che fanno parte dell'ecosistema.

Questa soluzione permette (ancora una volta) di gestire i dispositivi in maniera sicura. Le "cose" infatti mandano informazioni ad internet, ma da questa ricevono anche dei comandi. E se pensiamo che solo in Italia si stimano circa 93 milioni di oggetti connessi ad Internet, la blockchain può essere la soluzione per dare un'identità crittografica ai dispositivi IoT e assicurarci che il comando venga inviato in maniera univoca.

Siamo solo all'inizio di questa rivoluzione e non mancheremo di mostrarvi le potenzialità dell'IoV.

Boris Landoni

boris.landoni@elettronica.it



In copertina:
In questo numero vedremo come l'intelligenza artificiale ci aiuta a discriminare le intrusioni di persone non autorizzate.

ON LINE

elettronica.it

SI RINGRAZIA

Futura Elettronica
Maker Faire
Mouser

Elettronica In

Rivista mensile, anno XXVII n. 258

OTTOBRE 2021

EDITORIALE

Direttore Responsabile

Boris Landoni

boris.landoni@elettronica.it

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

redazione@elettronica.it

Grafica

Alessia Sfulcini

alessia.sfulcini@elettronica.it

Ufficio Pubblicità

Monica Premoli (0331-752668)

monica.premoli@elettronica.it

Ufficio Abbonamenti

Tel. 0331-752668

abbonati@elettronica.it

Direzione, Redazione, Pubblicità

FUTURA GROUP srl - Divisione Editoriale

via Adige 11 - 21013 Gallarate (VA)

Tel. 0331-752668

Abbonamenti

Annuo 10 numeri Euro 45,00

Annuo 10 numeri digitale Euro 39,00

Le richieste di abbonamento vanno inviate a:

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

Tel. 0331-752668

oppure tramite il sito

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

Distribuzione per l'Italia

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

via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

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

Tiratura
18.000 copie



Mensile
associato all'USPI,
Unione Stampa
Periodica Italiana

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

Prezzo di copertina Euro 6,00.

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

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

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

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

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

© 2021 Futura Group srl

Futura
Group
Edizioni

Contenuti

01

RUBRICHE

- 01 Editoriale
- 05 T&T Tips & Tricks
- 09 Embedded IoT
Digital Transformation of Things
- 13 Componenti & Sistemi
- 126 Scienza & Tecnologia

Didattica

Totem Robo
Board x4

35



19

ARTICOLI

STRUMENTAZIONE

19

Sonde differenziali per alta tensione per oscilloscopio

Vediamo cosa sono, come sono fatte e a cosa servono, le sonde differenziali per alta tensione per oscilloscopio.

LABORATORIO

27

Micro oscilloscopio

Indispensabile nel nostro laboratorio di elettronica, ha le funzioni basilari di uno strumento economico e la base dei tempi da 500 kps per misure in BF.

Progetto di Copertina



person: 1.0



ALLARME ANTI-INTRUSIONE CON AI

109

Strumentazione

Sonde differenziali
per alta tensione

19



Video

Lettura Green Pass
con OpenCV e Python

42



Laboratorio

Micro oscilloscopio

27



Raspberry Pi

Smart Mirror con Alexa

55



Arduino**RFID con RC522****69****Didattica****La conversione A/D veloce per Fishino32****77****DIDATTICA****35 Totem RoboBoard X4**

Impariamo a sviluppare applicazioni e a programmare con la RoboBoard X4 di Totem, la piattaforma per la robotica educativa dell'ecosistema Totem.

VIDEO**42 Lettura Green Pass con OpenCV e Python**

Leggiamo con una telecamera e validiamo il QR GreenPass mediante OpenCV e Python eseguiti su una Raspberry Pi.

RASPBERRY PI**55 Smart Mirror con Alexa**

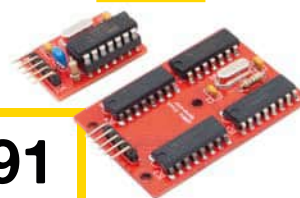
MirrorCoppo è uno specchio Smart realizzato con l'assistente vocale Amazon Alexa installato sulla Raspberry Pi, controllato da vocale o gesture. Seconda e ultima puntata.

ARDUINO**69 RFID con RC522**

Conosciamo e impariamo a utilizzare una scheda per l'identificazione di tag a RF, facilmente abbinabile a un microcontrollore e ad Arduino.

DIDATTICA**77 La conversione A/D veloce per Fishino32**

Campioniamo con un Fishino a 32 bit segnali analogici e calcoliamone lo spettro. Un OLED di piccole dimensioni ci mostrerà in tempo reale il risultato.

Laboratorio**Piatto per saldatura a rifusione****Laboratorio****Generatore 100Hz****91****Didattica****Conoscere e usare Node-RED****117****LABORATORIO****91 Generatore 100Hz**

Generiamo una frequenza stabile e precisa utilizzando e confrontando la soluzione a logica cablata e quella basata su microcontrollore.

LABORATORIO**99 Piatto per saldatura a rifusione**

Soluzione semplice ed economica per assemblare o riparare piccoli circuiti SMD nel nostro laboratorio.

RASPBERRY PI**109 Allarme anti-intrusione con AI**

Sistema di sicurezza basato sul riconoscimento d'immagine implementato eseguendo su una Raspberry Pi degli algoritmi di intelligenza artificiale.

DIDATTICA**117 Conoscere e usare Node-RED**

Iniziamo a scoprire un tool di flow-based programming, orientato all'IoT ed alla connettività, originariamente sviluppato dall' IBM Emerging Technology Services team e adesso parte della JS Foundation. Spiegheremo cos'è Node-RED ed inizieremo a prendere confidenza con l'ambiente e con l'editor grafico, mostrando poi alcuni esempi di utilizzo su Raspberry Pi. Prima Puntata

Gli strumenti di sviluppo in un unico posto

Migliaia di strumenti da centinaia
di produttori fidati



Scegli tra i prodotti della
nostra vasta selezione su
mouser.it/dev-tools



MOUSER
ELECTRONICS

Lunga vita alle batterie al litio

Per quanto siano molto prestanti e possano immagazzinare elevate quantità di energia elettrica in rapporto al loro peso e al volume occupato, le batterie Li-Ion e le LiPO vanno trattate con cautela; non bisogna esporle a elevate temperature (quindi al sole), non vanno schiacciate né perforate (perché potrebbero incendiarsi) e, sotto l'aspetto elettrico, non devono essere scaricate completamente. A riguardo notate che una cella viene considerata scarica quando la tensione ai suoi capi scende già a 3,4 V; ma la tensione limite sotto la quale ogni cella elementare non deve scendere è 2,7÷2,8V, altrimenti si danneggia irreversibilmente e una volta ricaricata non avrà più tutta la propria capacità originaria.

A piena carica, una cella al litio presenta una tensione di 4,2 V, che scende rapidamente a 3,7÷3,8 V dopo che è stato consumato circa 1/10 della carica disponibile; proseguendo la scarica, la tensione cala in modo abbastanza lineare fino a 3,4 V per il restante 80% della carica. Quando la tensione raggiunge i 3,4 V significa che è stato usato il 90% della carica disponibile; scaricando oltre l'accumulatore, la tensione cala molto rapidamente fino ai 3 V.

Di norma la batteria dovrebbe integrare un'elettronica che la sconnette (il metodo più semplice è adottare un comparatore di tensione con riferimento stabilizzato, che pilota una batteria di MOSFET a bassissima resistenza serie o dei teleruttori) dal carico o perlomeno riduce drasticamente l'assorbimento, quando la tensione per singola cella scende sotto i 3,4 V.

Nel caso dei dispositivi come quelli per i radiomodelli (ma non solo) a staccare la batteria quando la sua tensione è troppo bassa provvede il BEC (Battery Eliminator Circuit) che interrompe l'erogazione all'utiliz-

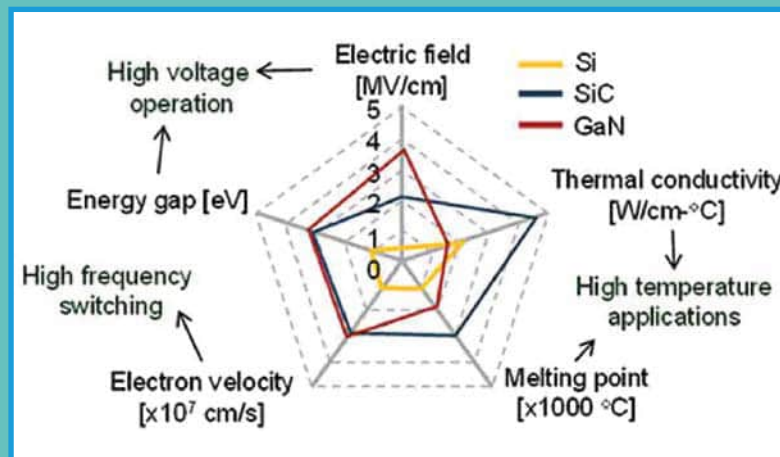
zatore, ovvero la riduce in termini di potenza erogata, per evitare una scarica eccessiva.

Il degrado nel tempo delle LiPO è leggermente più rapido di quello delle batterie agli ioni di litio, mentre l'autoscarica è dell'ordine del 10% nelle Li-Ion e del 5% nelle LiPO; la loro capacità utile si riduce progressivamente e abbastanza rapidamente con il passare del tempo; questo processo viene accelerato dal calore, se la temperatura delle batterie sale sopra i 40 °C. Per questa ragione,

è consigliabile tenere le batterie al freddo quando vengono stoccate nei magazzini.

Silicio e SiC: le differenze

Requisiti di potenza più elevati, rispetto delle normative e degli standard sul risparmio energetico e sulle problematiche EMI, stanno imponendo all'industria elettronica l'utilizzo di dispositivi di potenza a commutazione



↑ Confronto di massima tra le principali proprietà materiali del SiC e quelle dei materiali solidi Si e GaN. A confronto del Si, il SiC ha livelli più alti di cedimento critico, di conducibilità termica e una banda proibita più ampia.

per gli alimentatori, in considerazione della loro maggior efficienza e dell'intervallo di funzionamento più ampio. La concomitante necessità di riduzione dei costi e risparmio di spazio su PCB ha spinto a cercare un'alternativa ai classici MOSFET in silicio che ha preso forma nel carburo di silicio (SiC o Silicon Carbide). Si tratta di un semiconduttore di sintesi composto da carbonio e silicio, che può essere drogato come si fa con i semiconduttori naturali in modo da realizzare componenti attivi. Il principale campo di applicazione sono i MOSFET di potenza, perché i FET a base di carburo di silicio offrono molti vantaggi, in particolare per ciò che concerne l'efficienza, la maggiore affidabilità, la semplicità della gestione termica e l'ingombro ridotto a parità di potenza. Sotto l'aspetto della progettazione sono applicabili nell'intero spettro di potenza dei MOSFET tradizionali in silicio e la progettazione dei circuiti che li utilizzano non è tanto differente da quelle dei circuiti con semiconduttori in silicio. I FET in carburo di silicio sono MOSFET come quelli in silicio, la loro struttura fisica è simile e si tratta in entrambi i casi di dispositivi a tre terminali, connessi a source, drain e gate. Rispetto al silicio, il SiC ha tre proprietà elettriche fondamentali che differiscono in modo significativo, ciascuna delle quali offre vantaggi operativi:

- valori più alti della rigidità dielettrica (campo elettrico di breakdown), pari a circa 2,8 MV/cm, a fronte di 0,3MV/cm del silicio; il funzionamento a una determinata tensione nominale richiede pertanto uno strato molto più sottile di SiC, con la conseguente riduzione della resistenza nello stato On ($R_{ds(on)}$);
- maggiore conducibilità termica, che consente una densità di corrente più elevata e la riduzione dell'area del semiconduttore;
- banda proibita più larga (differenza di energia -in eV- tra la parte superiore della banda di valenza e il fondo della banda di conduzione nei semiconduttori e isolanti) che determina una corrente di dispersione più bassa alle alte temperature.

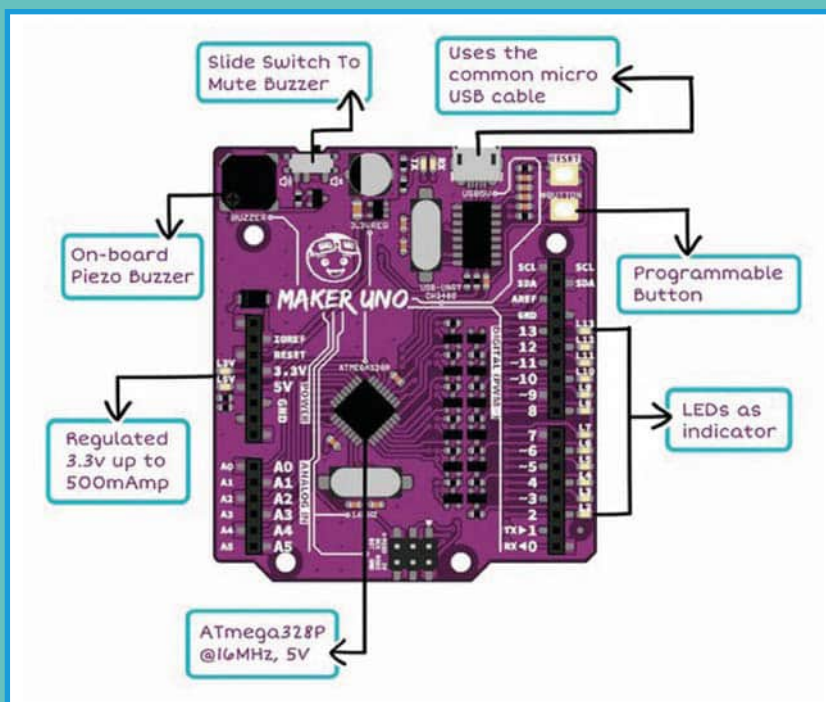
Di conseguenza, dispositivi in SiC sono in grado di commutare tensioni sino a dieci volte superiori rispetto a quelli in silicio e possono commutare dieci volte più velocemente con una resistenza nello stato On pari alla metà o meno a 25 °C. Al contempo, la loro capacità di funzionare a temperature di ben lunga superiori ai 200 °C rispetto ai 125 °C semplifica la progettazione e la gestione termica. Un elemento che distingue il tipico MOSFET di potenza in silicio da quello in SiC è il circuito di comando del gate. Per i FET in SiC, il driver deve avere altre fun-

zioni per poter minimizzare le perdite di conduzione e commutazione, nonché le perdite del gate, che comprendono l'energia di attivazione e disattivazione, l'effetto di Miller e i requisiti per la corrente di comando del gate. L'energia di disattivazione è una funzione della resistenza del gate e della tensione gate-sorgente nello stato off. Per ridurla, è necessario che la capacità di gate venga scaricata rapidamente e una maniera per fare ciò consiste nell'applicare, da parte del driver una tensione negativa al gate durante il tempo di disattivazione. Analogamente, l'energia di attivazione viene ridotta diminuendo la resistenza del gate.

Il driver di gate deve inoltre ridurre al minimo l'effetto di Miller e le sue conseguenze negative, cioè la possibile attivazione involontaria causata dalla capacità parassita in alcune circostanze e configurazioni applicative. L'attivazione indotta dall'effetto Miller aumenta l'energia di recupero inverso e si aggiunge alle perdite. Una soluzione consiste nel dotare il driver della funzione di protezione Miller Clamp, che controlla la corrente di comando durante la commutazione dello stadio di potenza.

Il driver deve fornire la corrente di sink e di source necessaria a valori di tensione idonei; i dispositivi in SiC in genere necessitano di tensioni di gate più alte (+20V) rispetto ai MOSFET in silicio per ridurre al minimo le perdite; possono anche richiedere una tensione di disattivazione del gate negativa compresa tra -2 e -6V. La tensione di gate necessaria viene stabilita con i soliti calcoli che si basano su carica del gate (Q_g), VDD, ID della corrente di drain, tensione gate-source e resistenza del gate ed è in genere nell'ordine di pochi ampere. Questa corrente deve avere valori nominali di sink e di source adeguati a una velocità di variazione commisurata alla velocità di commutazione del FET in SiC.

Tra le funzioni del driver di gate c'è anche quella di modellare e ridurre al minimo le correnti parassite della scheda e del dispositivo che possono produrre oscillazioni, sovraelevazioni di tensione/corrente e falsi positivi alle velocità di commutazione più alte offerte da questi dispositivi. I MOSFET in silicio presentano una caratteristica di corrente che smorza la sovraelevazione e la sovraoscillazione; i MOSFET in SiC no, perciò la sovraelevazione e la sovraoscillazione della tensione di drain possono essere elevate, dando origine a problemi. La diminuzione di queste correnti parassite richiede grande attenzione alle problematiche di layout, con la



↑ Layout della scheda Maker UNO.

riduzione al minimo delle lunghezze delle piste e la collocazione del driver il più vicino possibile al MOSFET. La riduzione della sovraoscillazione permette anche di ridurre le EMI (interferenze elettromagnetiche) associate alla commutazione ad alta velocità sia lato pilotaggio sia lato carico del dispositivo.

Per semplificare il lavoro del progettista, esistono circuiti integrati destinati al pilotaggio dei MOSFET in SiC. In molti progetti i gate driver e i FET in SiC devono possedere un isolamento galvanico dalla circuiteria di bassa tensione, che può essere implementato tramite tecniche di isolamento ottico, trasformatore di impulsi o isolamento capacitivo utilizzando componenti standard. L'isolamento in primo luogo ha come scopo la sicurezza degli utenti, che devono essere protetti dalle alte tensioni in caso di malfunzionamento del circuito, e in secondo luogo nelle varie topologie di circuito dove il MOSFET è intrinsecamente non a massa, come nelle configurazioni a ponte.

Il primo MOSEFT in SiC a scopo commerciale, il CMF20120D, è stato realizzato da Cree/Wolfspeed nel gennaio del 2011; aveva una V_{ds} di 1.200V e una corrente di drain di 98A, con 80 mΩ di resistenza nello stato On.

I dispositivi possono avere un pin Kelvin source che riduce l'effetto dei picchi di tensione nel circuito del gate; è quindi possibile applicare al gate tensioni più elevate ottenendo una commutazione dinamica più veloce, il che riduce le perdite di commutazione fino a 3,5 volte.

Makeruno: basso costo, alte prestazioni

Cloni della popolare Arduino, soprattutto della UNO, se ne trovano parecchi, ma uno molto interessante per le sue caratteristiche è quello chiamato Maker UNO; si tratta di una scheda di prototipazione compatibile con Arduino UNO Rev. 3, ma dotata di qualcosa in più. Prodotta dalla Cytron Technologies e basata sull'ATmega328P come la Arduino UNO, dispone di 12 LED on-board (pilotati internamente dai pin digitali dal 2 al pin 13) utilizzabili per funzioni personalizzate, 1 buzzer piezoelettrico (controllato dal pin 8) a bordo, 1 interruttore on/off per il buzzer e 1 pulsante programmabile (pin 2). La corrente erogabile ai dispositivi funzionanti a 3,3 V attraverso il pin dell'header è di ben 500 mA, il che rende più adatta di Arduino UNO (che eroga appena 50 mA) alle applicazioni che



↑ La nuova linea di calcolatrici con cui programmare in Python.

richiedono breakout board sensore e dispositivi vari alimentati a 3,3 volt (oggi sono la maggioranza). Il convertitore USB-seriale ATmega16u2 della MakerUNO è il CH340, più economico del classico FT232 della FTDI ma ugualmente prestante e compatibile USB 2.0. Il prospetto visibile in queste pagine propone un confronto fra Arduino UNO e MakerUNO. Maker UNO ha anche una versione avanzata chiamata Maker UNO plus, dove il convertitore USB/seriale non è più il chip CH340 ma l'FT231X. Sotto l'aspetto della programmazione, la scheda è compatibile con l'IDE Arduino, per il quale sono previsti i driver di terze parti e i driver USB scaricabili dalla pagina web del sito Cytron <https://www.cytron.io/p-maker-uno-simplifying-arduino-for-education>.

Imparare Python sulla calcolatrice

Texas Instruments (TI), il noto produttore di calcolatrici grafiche, ha pensato agli studenti dei nostri giorni proponendo una speciale calcolatrice che li aiuta a esplorare, visualizzare e comprendere meglio la matematica e le scienze, ma soprattutto che introduce al linguaggio di programmazione Python, tra i più diffusi e in più rapida crescita al mondo. Python è un linguaggio di programmazione molto potente ma al contempo abbordabile, adatto sia per i principianti che per i programmatori avanzati ed è ampiamente utilizzato da molte delle aziende tecnologiche leader a livello mondiale in aree come lo sviluppo di app e l'intelligenza artificiale. Aggiungendo Python alle calcolatrici che molti studenti conoscono e usano già a scuola, si può rendere la programmazione più accessibile a tutti, eliminando la necessità, per gli

insegnanti, di riservare laboratori informatici separati per insegnare queste importanti abilità. Con Python, gli studenti imparano a programmare ed anche a pensare in modo critico e creativo per risolvere i problemi del mondo reale.

Le calcolatrici di TI facilitano l'apprendimento della matematica perché consentono di studiare più intuitivamente le funzioni, come ad esempio la Serie di Fibonacci. La calcolatrice dispone di un Editor che permette di visualizzare funzioni e risultato; ha anche uno script manager, offre una panoramica dei file Python script archiviati nella calcolatrice grafica e permette di creare nuovi script o editarli e avviarli.

La TI 84 Plus CE-T Python Edition permette inoltre di scrivere codice e fare esperimenti con lo strumento di programmazione TI-Innovator Hub e il veicolo robotizzato TI-Innovator Rover, da utilizzare in diversi progetti STEM. È anche possibile scrivere ed eseguire programmi Python per micro:bit direttamente dalla calcolatrice, trasferendoli poi via USB.

Lo schermo a colori con qualità grafica facilita la lettura dei programmi, distinguendo le parti del codice per colore e indentazione.

Web Forum e supporto tecnico

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

www.elettronicaain.it/webforum

La TV cambia? Non cambiarla!

Arriva la nuova TV digitale, acquista il decoder senza cambiare la TV!

Decoder TV digitale terrestre DVBT, DVBT2, H.265/HEVC



cod. GS240T2
€ 29,00

**NUOVO
MODELLO**

DVBT/T2
TERRESTRIAL

**H.265
HEVC**

**FULL HD
1080**

USB

HDMI

mpeg
VIDEO

XVID
VIDEO

GoSat GS240T2 è un decoder digitale terrestre di nuova generazione compatibile con il nuovo standard di trasmissione DVBT, DVBT2 H.265/HEVC. L'apparecchio può ricevere comunque trasmissioni nei vecchi standard DVB-T H.264 e MPEG-4. Il dispositivo offre una perfetta qualità delle immagini in Full HD 1080p. Grazie alla porta USB, disponibile sulla parte frontale, e alla funzione PVR (Personal Video Recorder), permette la registrazione di programmi TV e radio su chiavetta USB o su hard disk esterno. Può essere inoltre connesso a internet tramite il connettore RJ45 per utilizzare servizi come YouTube e guardare canali IPTV (liste con estensioni M3U). Il telecomando in dotazione, oltre ad avere tutte le funzioni per il controllo del ricevitore, ha la funzione di apprendimento e può essere utilizzato anche per altri dispositivi (TV, ecc.). Il pannello frontale ha un display che mostra l'ora o il numero del programma selezionato.



Il tuo telecomando TV è rotto?

Sostituiscilo con i telecomandi universali a infrarossi compatibili con le TV delle maggiori marche prodotte dal 2000 ad oggi. Dispongono di tutte le funzioni del telecomando originale. Subito utilizzabili!

€ 5,90
OGNI PEZZO

**PRONTI
ALL'USO!**



Cod. TVIR-SAMSUNG



Cod. TVIR-SONY



Cod. TVIR-PHILIPS



Cod. TVIR-LG



Cod. TVIR-PANASONIC

DA PROGRAMMARE

Per comandare fino a due dispositivi audio/video.

€ 9,00

cod. TVFREEDOM2



DISPONIBILE ANCHE 1 in 1
cod. TVFREEDOM1

€ 8,50

I sistemi di elaborazione edge necessitano di dispositivi programmabili compatti, a basso consumo energetico e con ingombro termico sufficientemente ridotto da eliminare le ventole e altre misure di mitigazione del calore, fornendo allo stesso tempo elevata potenza di calcolo. Microchip Technology ha risolto questa sfida dimezzando il consumo di energia statica per i suoi dispositivi FPGA (Field Programmable Gate Arrays) a larghezza di banda media e System-on-Chip (SoC) FPGA e offrendo il minimo ingombro termico e le migliori prestazioni e potenza di calcolo. Le recenti Microchip PolarFire FPGA a bassa densità (MPF050T) e PolarFire SoC (MPFS025T) superano le metriche prestazioni/potenza di qualsiasi FPGA a bassa densità o SoC FPGA sul mercato, con tessu-

› Mid-Range FPGA per Edge Computing



to FPGA veloce e architettura RISC-V, con 2 MB di cache L2 e supporto di memoria Low-Power DDR4 (LPDDR4). Il portafoglio è stato recentemente ampliato dal SoC RISC-V multi-core da 25K elementi logici e un FPGA da 50K elementi

logici. Sono ideali per applicazioni di visione integrata intelligente a bassa potenza e sistemi automobilistici, di automazione industriale, di comunicazione, di difesa e IoT. I PolarFire sono integrati da una suite di dispositivi Microchip per soluzioni di sistemi completi per applicazioni tra cui visione integrata intelligente, apprendimento automatico, sicurezza, settore aerospaziale e difesa e calcolo integrato. Forniscono inoltre soluzioni plug-and-play per i progetti di alimentazione e temporizzazione. Gli FPGA SmartFusion 2 e PolarFire forniscono il miglior equilibrio tra fattore di forma, efficienza e risorse di elaborazione, a supporto di algoritmi incorporati come compensazione senza otturatore e miglioramento dell'immagine.

www.microchip.com

› Inmarsat presenta Elera, la rete globale IoT

Dopo il lancio di Orchestra, la prima infrastruttura al mondo a integrare Geo, Leo e 5G, Inmarsat (creatore della prima rete di telefonia satellitare) punta ad accrescere l'uso della connettività narrowband nel mondo dell'industria, dell'agricoltura e dei trasporti. Lo farà con una rete globale narrowband per gestire la rapida evoluzione dell'Internet of Things (IoT) e per la connettività nei settori dell'aviazione (anche degli UAV, gli aerei senza equipaggio...), del trasporto marittimo ecc. chiamata Elera.

In dettaglio Elera, farà da trampolino allo sviluppo di casi d'uso, facendo leva sulla banda di comunicazione L, su terminali più piccoli e a basso costo e sulla costellazione di satelliti Inmarsat-6, la prima (I-6 F1) il cui lancio è stato programmato entro la fine dell'anno.

Le capacità della nascente Elera, combinate con lo spettro di Inmarsat e la posizione orbitale delle reti satellitari,



ri, amplieranno l'ambito applicativo dell'IoT.

Le recenti innovazioni in cui l'azienda sta investendo includono la tecnologia di gestione dello spettro RF della rete satellitare, mirata a fornire velocità di comunicazione dati in banda L fino a 1,7 Mbps.

Le nuove capacità di gestione dello spettro (Carrier Aggregation) incorporate nella rete Elera forniranno le velocità più elevate disponibili a livello globale ai clienti in banda L, superando di gran lunga le capacità di qualsiasi altra rete in banda L mondiale.

Le prove sul campo nel settore dell'aviazione commerciale dovrebbero iniziare nel 2022 e la tecnologia sarà implementata su una gamma di applicazioni specifiche. Elera fornirà anche un'importante estensione al portafoglio di dispositivi abilitati alla comunicazione vocale di Inmarsat, portando nuove funzionalità e innovazioni a centinaia di migliaia di clienti; non va infatti dimenticato che Inmarsat nacque come telefonia satellitare (portatile) implementata con una rete di 4 satelliti.

www.inmarsat.com

› Western Digital OptiNAND: HD per l'IoT

Western Digital ha presentato una nuova architettura basata su tecnologia Flash per HDD, chiamata OptiNAND, che sfrutta la tecnologia flash iNAND per ottimizzare gli hard disk meccanici. Grazie all'impiego di attuatori a triplo stadio (TSA) e alla tecnologia HelioSeal, le unità basate su OptiNAND saranno in grado di offrire capacità sino a 2,2TB per piatto magnetico, ampliando ulteriormente lo spazio offerto dalla precedente tecnologia ePMR.

Western Digital ha una lunga tradizione di innovazione nell'architettura degli hard drive: ad esempio HGST (ora parte di Western Digital) fu la prima nel 2013 a implementare le unità HDD a elio. Guidata dalla crescita dell'Intelligenza

Artificiale, del Machine Learning, della blockchain, dell'IoT, dei sensori e molto altro, l'innovazione nello storage è necessaria per memorizzare e proteggere la mole crescente di dati, i cosiddetti Big Data, soprattutto su larga scala. Come unica azienda che produce sia flash che HDD, Western Digital può sfruttare in modo unico le proprie potenzialità interne per estendere la curva di densità areale delle unità ePMR per le esigenze future, aiutando a soddisfare le crescenti necessità di un'economia digitale.

Le caratteristiche chiave della tecnologia OptiNAND sono:

- capacità, perché l'unità funziona in maniera più smart, con migliorati algoritmi del firmware che si avvantaggiano dei

› Prevenire le perdite di gas con LoRaWAN

Semtech Corporation, leader nelle soluzioni wireless per l'IoT basate sul protocollo LoRa (del quale è stata promotrice), ha annunciato la collaborazione con eLichens, progettista di avvolta-CH4, un nuovo rilevatore di fughe di gas connesso tramite LoRaWAN. Basato sulla tecnologia di rilevamento del gas brevettata da eLichens, avvolta è in grado di rilevare perdite di gas naturale (metano) con precisione, non ha deriva nel tempo ed è stato approvato dai laboratori di ricerca sui gas. Si stima che un quarto di tutte le perdite di gas

dai gasdotti e dalle infrastrutture del gas naturale fuoriesca vicino ai contatori del gas. Ecco perché un rivelatore, collegato in rete wireless LoRaWAN e capace di comunicare le anomalie in tempo reale, è di vitale importanza. La combinazione della tecnologia dei sensori NDIR eLichens e LoRaWAN consente la conformità diretta con gli standard UL1484 e EN50194-1. Il rilevatore di gas avvolta di eLichens in esecuzione su LoRaWAN presenta un'autonomia della batteria di oltre 10 anni, grazie all'impiego del sensore

di gas a bassissimo consumo energetico e all'integrazione del dispositivo Semtech LoRa e dello standard LoRaWAN. Il fatto di non richiedere la ricalibrazione e la lunga autonomia fornita dal sensore eLichens lo rendono ideale per applicazioni di rilevamento di fughe di gas nel mercato sia residenziale che industriale.

Sfruttando l'aumento dei sensori connessi e delle reti wireless a bassa potenza, le società di servizi del gas possono ridurre significativamente le perdite di gas sulla loro rete di distribuzione del gas.

Il nuovo rilevatore di fughe di gas di eLichens è in grado di rilevare e monitorare selettivamente (grazie alla tecnologia di rilevamento NDIR di eLichens) bassi livelli di perdite di metano, prendendo così le decisioni appropriate in linea con la natura della perdita. L'utilizzo di LoRa e LoRaWAN mostra le potenzialità delle tecnologie wireless IoT al servizio di un pianeta più intelligente e, in definitiva, più sicuro per tutti. Creata nel dicembre 2014, eLichens è una startup che sviluppa e commercializza soluzioni complete di sensori/dati/servizi per i mercati delle utility del gas, dell'industria, delle smart city, delle smart home e dell'IoT.

<https://www.elichens.com/>



metadati estesi, scaricati sulla tecnologia iNAND. Questo consente più tracce per pollice (TPI) aumentando di conseguenza la densità areale.

- latenza dell'unità migliorata, grazie alle proprietarie ottimizzazioni a livello del firmware dell'unità che prevedono la richiesta di un minor numero di refresh delle interferenze delle tracce adiacenti (ATI) e la riduzione della necessità di flush della cache di scrittura nella modalità abilitata alla scrittura;
- può essere conservata una quantità di dati superiore di quasi 50 volte nel caso di uno scenario di spegnimento di emergenza (EPO) grazie all'offerta di soluzioni integrate verticalmente, nello sviluppo, nella progettazione, nel test e nella qualificazione delle unità flash-enhanced.

www.westerndigital.com



› Maxim e Xailient per il rilevamento edge di volti

Il microcontrollore AI MAX78000 di Maxim Integrated e la rete neurale Detectum di Xailient rilevano e localizzano i volti in video e immagini a soli 12 ms per inferenza. Xailient è un'azienda focalizzata sull'intelligenza artificiale (AI) per l'edge IoT e congiuntamente a Maxim ha annunciato che il MAX78000, microcontrollore di rete neurale, riesce a rilevare e localizzare i volti in video e immagini utilizzando la rete neurale proprietaria Detectum di Xailient. Questa rete neurale consuma un'energia 250 volte inferiore (di soli 280 microJoule) rispetto alle soluzioni embedded convenzionali e a 12 ms per inferenza; funziona in tempo reale ed è più veloce della soluzione di rilevamento facciale più efficiente disponibile per l'edge IoT. La velocità è determinante per l'AI, perché con un'inferenza più rapida è possibile reagire in tempo reale o mediare rapidamente più inferenze per migliorare la precisione. I sistemi di intelligenza artificiale alimentati a batteria che richiedono il rilevamento del volto, come le telecamere domestiche, le telecamere di sicurezza intelligenti di livello industriale e le soluzioni di vendita al dettaglio, necessitano una soluzione a bassa potenza per fornire il funzionamento

più lungo possibile tra una ricarica e l'altra. Oltre a supportare applicazioni standalone, il microcontrollore di Maxim Integrated, abbinato alla rete neurale di Xailient, migliora l'efficienza energetica complessiva e la durata della batteria delle applicazioni ibride edge/cloud che utilizzano una modalità di "ascolto" a bassa potenza che attiva i sistemi più complessi quando viene rilevato un volto.

La rete neurale Detectum di Xailient include tecnologie di messa a fuoco, zoom e parole di attivazione visive per rilevare e localizzare i volti in video e immagini a velocità 76 volte superiori rispetto alle soluzioni software convenzionali, con una precisione simile o migliore. Inoltre, la rete flessibile può

essere estesa ad applicazioni diverse dal riconoscimento facciale, come l'inventario e il monitoraggio del bestiame, l'occupazione dei parcheggi, i livelli di inventario e altro ancora.

La rete neurale Xailient ottimizza l'efficienza computazionale e le modalità di sospensione flessibili a basso consumo offerte dal microcontrollore MAX78000. Insieme, le soluzioni consentono anni di autonomia di funzionamento delle applicazioni edge/cloud ibride alimentate a batteria.

Con la rete neurale Xailient Detectum, il MAX78000, oltre a vedere i volti nell'immagine o nel video permette di determinare dove si trovano quei volti nel campo visivo dell'immagine.

www.maximintegrated.com



CON IL PATROCINIO DI

ROMA



CON IL SUPPORTO DI



Ministero degli Affari Esteri
e della Cooperazione Internazionale

ITA

ITALIAN TRADE AGENCY
ICE - Agenzia per la promozione all'estero e
l'internazionalizzazione delle imprese italiane

IN PARTNERSHIP CON

INAIL



REGIONE
LAZIO

IN COLLABORAZIONE CON



UNIONCAMERE
LAZIO

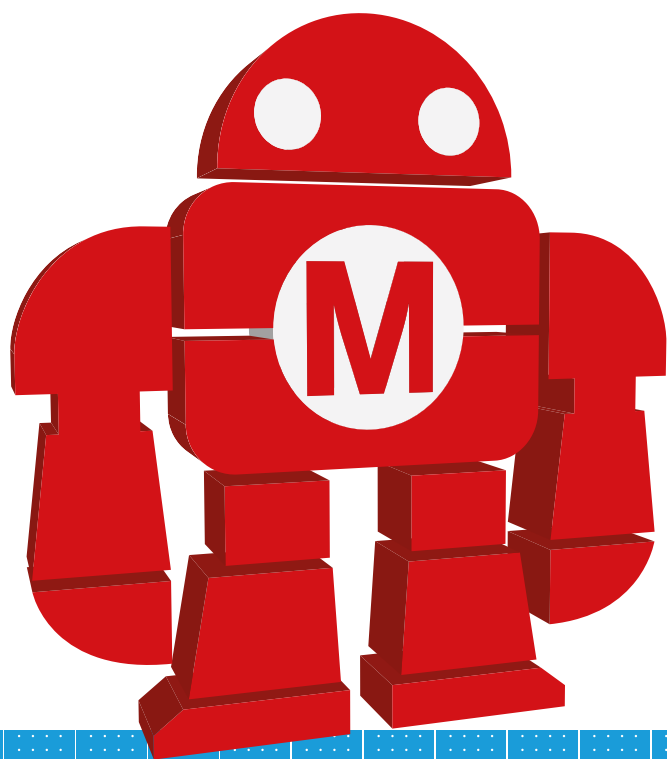
CAMERE DI COMMERCIO DEL LAZIO

ROME

Maker Faire

THE EUROPEAN EDITION

IL PIÙ GRANDE EVENTO EUROPEO
SULL'INNOVAZIONE



**8 - 10
OTTOBRE**



#MFR2021

MAKERFAIREROME.EU

MAIN PARTNER



GOLD PARTNER



Five Years Out

acea



SONY



MEDIA PARTNER



PROMOSSO
E ORGANIZZATO DA



Camera di Commercio
Roma

MOSFET di potenza a canale N a basso picco da Toshiba

Toshiba ha lanciato il TPHR7404PU, un MOSFET a canale N da 40V che utilizza il processo U-MOSIX-H di ultima generazione, caratterizzato dal basso picco che riduce le sovraelevazioni nelle applicazioni a picco. Ciò rende il dispositivo ideale per la rettifica sincrona sul lato secondario nelle applicazioni di alimentazione a commutazione, grazie al suo contributo alla riduzione delle EMI. Questo risultato è ottenuto mantenendo una resistenza di accensione estremamente bassa, pari a soli 0,74mΩ (max) con una VGS di 10 V. Una tensione di soglia del gate compresa fra 2V e 3V ($I_d = 1mA$) aiuta ad evitare l'accensione indesiderata, ad esempio negli strumenti alimentati a batteria.

Il TPHR7404PU presenta una struttura innovativa che utilizza una rete di snubber parassita capace di limitare il rumore e l'oscillazione durante la commuta-



zione. Ciò si traduce in una tensione di picco inferiore durante la commutazione e in una riduzione dell'oscillazione del segnale, a vantaggio di DC/DC, driver per motori e regolatori di tensione switching.

Il MOSFET è disponibile in un package

SOP Advance da 5 x 6 mm, e le sue dimensioni è disponibile e la resistenza termica da canale a case estremamente ridotta di 0,71°C/W lo si rende adatto alle soluzioni di alimentazione compatte e ottimizzate per l'efficienza.

www.toshiba.com

Moduli di alimentazione aerospace



Nella riduzione delle emissioni dei velivoli rientrano i sistemi elettrici che sostituiscono la pneumatica e l'idraulica, alimentando dagli alternatori di bordo agli attuatori e alle unità di alimentazione ausiliarie (APU). Microchip ha annunciato lo sviluppo con Clean Sky, dei primi moduli di alimentazione baseless per il settore aerospaziale che consentono una conversione di potenza e un controllo degli aziona-

menti più efficiente; si tratta della famiglia BL1, BL2 e BL3, conforme RTCA DO-160G, comprendente MOSFET SiC e Schottky Barrier Diodes da 100W a più di 10 kW. Questi moduli di alimentazione sono disponibili con tensione da 600V a 1.200V per MOSFET e IGBT e 1.600V per i raddrizzatori. Microchip lavora anche sulla limitazione dell'obsolescenza dei componenti.

www.microchip.com

Xilinx Versal HBM per i big data

Xilinx ha introdotto la piattaforma di accelerazione di calcolo adattiva (ACAP) Versal HBM, che supporta la convergenza di memoria veloce, connettività sicura ed elaborazione adattabile in un'unica piattaforma. Gli ACAP Versal HBM integrano la DRAM HBM2e più avanzata, fornendo 820 GB/s di throughput e 32 GB di capacità per una larghezza di banda di memoria 8 volte maggiore e una potenza inferiore rispetto alle implementazioni DDR5.1.

I dispositivi Versal HBM incorporano core di rete ottimizzati dal punto di vista energetico per una larghezza di banda elevata e una connettività sicura. La serie Versal HBM offre 5,6 Tb/s di larghezza di banda seriale con rice trasmettitori PAM4 da 112 Gb/s, 2,4 Tb/s di larghezza di banda Ethernet scalabile, 1,2 Tb/s di throughput di crittografia del-

la velocità di linea, 600 Gb/s di connettività Interlaken e 1,5 Tb/s di larghezza di banda PCIe Gen5 con DMA integrato, che supporta sia CCIX che CXL. Questo ampio set di IP rinforzati fornisce connettività di rete multi-terabit per un'ampia gamma di protocolli, velocità dati e standard ottici. La serie Versal HBM integra motori adattabili per il parallelismo hardware a bassa latenza, motori DSP per l'inferenza AI e l'elaborazione del segnale e motori scalari per calcolo, gestione della piattaforma e avvio e configurazione sicuri.

www.xilinx.com



Keysight per test complessi sul 5G



Vivo ha scelto l'emulatore di canale a RF PROPSIM F64 di Keysight per eseguire test over-the-air (OTA) MIMO in un ambiente di laboratorio sotto un vasta gamma di scenari di mobilità reali come quelli rurali, urbani e indoor. Le soluzioni di emulazione del canale

PROPSIM di Keysight consentono di testare le prestazioni dei dispositivi 5G in condizioni di canale reali secondo le ultime specifiche radio 5G (NR) definite da 3GPP. PROPSIM si integra con la piattaforma di test wireless UXM 5G per fornire scenari di test MIMO OTA

definiti da CTIA nella gamma di frequenza 1 (FR1). Combinando le capacità di emulazione di canale e rete sono disponibili funzionalità di test delle prestazioni, tra cui la segnalazione del protocollo end-to-end dello stack completo.

www.keysight.com

Rivoluzione della realtà aumentata con il nuovo scanner MEMS di Infineon per occhiali e display head-up



Infineon Technologies ha sviluppato una nuova soluzione di scanner MEMS che comprende un mirror MEMS e un driver MEMS. Le sue dimensioni miniaturizzate e il basso consumo energetico sono la base per rendere le soluzioni di realtà aumentata (AR) più ampiamente disponibili per le applicazioni consumer, come i dispositivi indossabili e gli head-up display automobilistici.

Le soluzioni di realtà aumentata arricchiscono gli ambienti reali con preziosi dati digitali e aiutano le persone a muoversi in modo più comodo e sicuro nella vita quotidiana, per le attività di lavoro e di svago, ma soprattutto per strada. La sovrapposizione di informazioni preziose, dalla navigazione del percorso ai sistemi di assistenza alla guida, sull'intero parabrezza dell'auto, invece che su una piccola area davanti al conducente, possono migliorare la sicurezza e la comodità di guida. Il chipset per scanner MEMS di Infineon è dotato di un innova-



tivo specchio inclinabile che pone le basi per una nuova generazione di proiettori con scanner a laser (LBS). I sistemi HUD convenzionali possono avere un volume ottico superiore a 30 litri, ma offrono un campo visivo modesto. Al contrario, gli scanner a raggio laser Infineon consentono sistemi HUD con un volume ottico

minimo, in modo che possano essere integrati anche nel cruscotto più piccolo. Il chipset Infineon MEMS consente la progettazione di un micro-proiettore AR leggero e può essere esteticamente integrato in occhiali da vista e sportivi, grazie al basso consumo energetico.

www.infineon.com

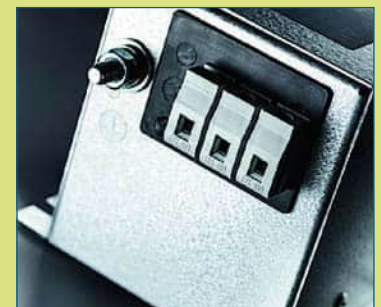
FMBC EP: il filtro di rete trifase a doppio stadio

Schurter introduce la famiglia di filtri a doppio stadio FMBC EP per sistemi trifase a ingombro limitato, adatti ai moderni quadri elettrici industriali. Il filtro a doppio stadio e l'utilizzo di componenti di altissima qualità (condensatori a film di generose dimensioni, induttanze con nuclei altamente permeabili) forniscono

eccezionale attenuazione. La famiglia di filtri a doppio stadio è adatta anche per applicazioni con elevate emissioni EMC, tipiche dell'industria. Le applicazioni includono macchine e sistemi per la produzione automatica, azionamenti di motori e macchinari per impianti industriali, convertitori di frequenza per il controllo motore.

La serie FMBC EP è dotata di terminali a vite e flangia metallica per il collegamento di terra. I filtri sono progettati per correnti da 16 a 230 A a una temperatura ambiente di 50 °C. Hanno sia l'approvazione ENEC che cURus e sono adatti fino 760 Vca, con correnti di dispersione <4 mA.

www.schurter.ch



PMIC per RZ/G2L e V2L da Renesas



Renesas presenta il suo nuovo PMIC (Power Management IC) RAA215300 ottimizzato per integrare i microprocessori RZ/V2L e RZ/G2L progettati per applicazioni abilitate per l'intelligenza artificiale. Il nuovo PMIC include nove canali, un caricabatterie integrato e un orologio in tempo reale; include sei regolatori buck VTT, tre LDO e un caricatore per duperca e batterie Li-ion a bottone. Supporta memorie DDR4, DDR4L, DDR3 e DDR3L con rail VREF, VTT e VPP dedicati. L'alto livello di integrazione aumenta l'affidabilità del sistema, grazie al minor numero di componenti necessari.

Il RAA215300 si accoppia con gli MPU RZ/G2L, RZ/V2L e altre soluzioni Renesas, in una combinazione per un SoM System-on-Module SMARC (Smart Mobility ARCHitecture) scalabile con design AI. Oltre a MPU e PMIC, questa combinazione include controller di alimentazione, un controller USB PD e un RTC. Renesas offre oltre 250 combinazioni che consistono in dispositivi analogici complementari, alimentazione, temporizzazione ed elaborazione incorporata e forniscono un'architettura facile da usare. Dotato di sequencer di alimentazione integrato e supporto per controlli on/off del regolatore DC/DC esterno, il PMIC sopporta temperature operative ambiente da -40 °C a 105 °C e implementa lo Spread Spectrum per ridurre le interferenze elettromagnetiche nelle applicazioni RF.

www.renesas.com

Panasonic utilizza il SoC Nordic

Panasonic ha scelto il SoC multiprotocollo nRF52820 Bluetooth 5.2/Bluetooth Low Energy (Bluetooth LE) di Nordic Semiconductor come core del proprio modulo PAN1781. Il potente processore Arm Cortex M4 a 64 MHz e 32 bit del SoC Nordic e l'allocazione di memoria RAM da 256 kB e RAM da 32 kB garantiscono che il modulo possa supportare un'ampia varietà di applicazioni Industrial Internet of Things (IIoT). Il PAN1781 è completamente certificato per CE RED, FCC e ISED ed è qualificato per l'intervallo di temperatura di esercizio industriale da -40° a 85° C. L'nRF52820 consente di sfruttare l'elevata velocità di trasmissione (2 Mbps) e il lungo raggio del



Bluetooth 5 e il rilevamento della direzione del Bluetooth 5.1 (incluso il supporto per l'angolo di arrivo (AoA) e l'angolo di partenza (tecnologia AoD) in applicazioni IoT. Il modulo supporta anche i protocolli Bluetooth mesh, Thread e Zigbee. Il PAN1781 è adatto ad applicazioni alimentate a batteria, grazie alla sua bassa corrente di idle di 1,5 µA. Il modulo integra anche un oscillatore esterno a quarzo a 32kHz e una radio multiprotocollo a 2,4 GHz con PA on-chip.

www.nordicsemi.com

OLED controller per auto da Melexis

Melexis ha sviluppato il controller OLED intelligente MLX81130, che integra tutti i componenti attivi necessari per realizzare applicazioni di illuminazione combinata puntiforme (LED) e area (OLED) all'interno e all'esterno del veicolo. Il chip può pilotare 25 LED/OLED e supporta l'interfaccia Melexis Light Bus (MeLiBu) ed è particolarmente adatto per luci posteriori combinate (RCL) o piccoli display OLED.

Melexis, una società globale di ingegneria microelettronica, ha introdotto l'MLX81130, un controller intelligente altamente integrato che consente a LED e OLED di essere pilotati da un singolo dispositivo. Progettato per supportare l'integrazione del sistema ASIL B, l'MLX81130 in un pacchetto QFN 5x5 dispone di 25 sorgenti di corrente costante programmabili. È stato progettato per funzionare nelle configurazioni single-master e multiple-slave.

Gli ingegneri automotive sanno che l'utilizzo dei LED insieme agli OLED richiede in genere due tipi di driver, con due controller e forse due interi sottosistemi che non sono facilmente integrabili. L'MLX81130 supera

questo problema, grazie alle uscite programmabili in modo indipendente, che possono pilotare sia LED che OLED. Ogni dispositivo può pilotare fino a 25 LED indipendenti o 8 LED RGB, supportando nel contempo l'intervallo di soglie esteso necessario per gli OLED. L'MLX81130 è stato progettato per supportare configurazioni multiple-slave, il che significa che il numero totale di LED/OLED che possono essere controllati da un Master è quasi illimitato.

Ogni uscita a corrente costante è controllata tramite il microcontrollore pipeline MLX16 FX integrato a 16 bit, supportato da un coprocessore matematico a 32 bit e 32 KB di Flash. Sono forniti anche due timer watchdog con clock indipendenti e un ADC a 10 bit. Le uscite possono essere controllate tramite PWM.

Queste correnti di uscita sono programmabili in step di 1mA tra 0mA e 15mA, quindi step di 2mA fino a 30mA per uscita, rispondendo alle esigenze del mercato. Per una corrente di uscita ancora maggiore, è possibile collegare in combinazione più uscite MLX81130.

www.melexis.com





ABBONAMENTO
ANNUALE



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



Visita: www.elettronica.in/abbonamenti



ABBONAMENTO ANNUALE PER RIVISTA DIGITALE

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

COME ABBONARSI

→ **on-line**

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

→ **e-mail**

scrivendo a
abbonati@elettronica.in

→ **telefono**

telefonando a
+39-0331-752668

ABBONAMENTO ANNUALE PER RIVISTA **CARTACEA**

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

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

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

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



INDISPENSABILE IN LABORATORIO!



L'alimentatore è lo strumento più importante di ogni laboratorio elettronico da cui si esige prima di tutto la possibilità di regolare con buona risoluzione la tensione e la corrente di uscita, requisiti soddisfatti da questi nuovi alimentatori. Mantengono i valori impostati in modo affidabile e indipendentemente dalle fluttuazioni della tensione di alimentazione o del carico collegato.



DISPONIBILE ANCHE NELLE VERSIONI:

DA 0-30 VDC / 0-10 A

COD. AL3010 € 106,00

DA 0-60 VDC / 0-5 A

COD. AL6005 € 106,00

€ 78,00

Cod. AL3005N



ALIMENTATORE LINEARE CON USCITA REGOLABILE 0-30 VDC / 0-5 A

Alimentazione: 110/220 VAC, 50/60 Hz

Uscita duale regolabile:

- Tensione: da 0 a 30 VDC

- Corrente: da 0 a 5 A

Risoluzione della tensione: 10/100 mV

Risoluzione della corrente: 1/10 mA

Precisione visualizzazione della tensione: $\pm 1\%$ +1 cifra

Precisione visualizzazione della corrente: $\pm 1\%$ +2 cifre

Dimensioni: 220x165x85 mm

Peso: 1,350 kg



Prezzi IVA inclusa.

SONDE DIFFERENZIALI PER ALTA TENSIONE PER OSCILLOSCOPIO

dell'Ing. **VINCENZO
MENDOLA**

Vediamo cosa sono, come sono fatte e a cosa servono, le sonde differenziali per alta tensione per oscilloscopio.



In ogni laboratorio elettronico che si rispetti, non può mancare l'oscilloscopio, strumento di misura principe tra gli quelli utilizzati dal progettista elettronico, per via della sua grande versatilità d'impiego per l'analisi e la verifica del funzionamento dei dispositivi elettronici. A volte impiegato fin dalle primissime fasi della progettazione elettronica, l'oscilloscopio è comunemente usato da chi si occupa di ricerca e sviluppo per testare, sperimentare e verificare la bontà e la fattibilità di nuovi prodotti e dispositivi, durante la progettazione per l'ottimizzazione di alcune parti

di circuito e per il test dei primi prototipi, durante la produzione di apparecchiature elettroniche viene utilizzato per la taratura di specifiche sezioni di sistemi e macchine ed infine dagli addetti al service come fondamentale ausilio nella ricerca guasti. Oggi è cosa abbastanza comune per l'appassionato di elettronica poter acquistare un oscilloscopio di buona qualità, con una banda passante di 200 MHz a 2 o 4 canali, già a partire da 500 euro o persino meno, con prestazioni una volta disponibili solo agli strumenti di alta gamma, come ad esempio la risoluzione minima dell'amplificazione verticale impostabile a 500uV/div, cosa possibile grazie all'uso di front end a basso rumore, inferiore al valore comunemente presente in questa categoria di oscilloscopi, tutto questo unitamente alla base dei tempi impostabile fino a 2nS/div, configurazione che permette di visualizzare nel dettaglio anche i fronti più ripidi delle forme d'onda. Un oscilloscopio con le caratteristiche appena menzionate, con una banda passante di 200 MHz e 4 canali è disponibile a poco meno di 500 euro su www.futurashop.it (cod. 6072-DSO204). Con poco più di 500 euro è possibile acquistare uno strumento sempre da 200MHz, 2 canali, che integra un generatore di funzioni arbitrarie (disponibile su www.futurashop.it cod. 6072-DSOFG202).

SONDE PER OSCILLOSCOPI DI TIPO PASSIVO E DI TIPO ATTIVO

Gli oscilloscopi sono normalmente venduti con delle sonde di misura di tipo passivo, con banda passante adeguata a quella dello strumento. Nel caso si volessero utilizzare, per qualsivoglia motivo, delle sonde differenti da quelle fornite, è molto importante fare attenzione alle loro caratteristiche durante la scelta, in modo da utilizzare solamente quelle caratterizzate da una banda passante adeguata allo strumento a cui verranno connesse o perlomeno sufficientemente ampia in relazione alla specifica misura che si vuole effettuare; questo per non incorrere in grossolani errori dovuti all'inadeguatezza della sonda che falserebbe la misura a causa di un'attenuazione indesiderata del segnale applicato all'ingresso dell'oscilloscopio. Si ricorda a tal proposito che le sonde fornite dal costruttore insieme allo strumento sono sempre adatte allo stesso e a volte presentano persino una banda passante di gran lunga superiore a quella dello oscilloscopio stesso, magari perché nella stessa famiglia di cui fa parte lo strumento ci sono modelli con caratteristiche superiori oppure a volte questo viene fatto perché la banda passante

dello strumento stesso è espandibile tramite una chiave software che viene fornita e venduta come fosse un miglioramento "fisico" dell'elettronica che invece rimane invariata e viene semplicemente "sbloccata"; in alcuni casi persino il numero di canali od opzioni come il generatore di funzioni arbitrarie integrato sono già presenti ma non sono attive finché non vengono acquistate ed abilitate. Un'altra grandezza caratteristica delle sonde passive è costituita dal loro fattore di attenuazione: spesso lo strumento è corredato da una coppia di sonde (o due coppie nel caso di strumenti a 4 canali) con un'attenuazione impostabile a 1:1 o 10:1, semplicemente agendo manualmente su di un selettore a slitta (in **Fig. 1** è mostrato un esempio di sonda passiva ad attenuazione fissa 10:1 da 500 MHz). Nel caso 1:1 non si ha nessuna attenuazione ed il valore di tensione in uscita dal BNC della sonda è lo stesso presente sul lato d'ingresso, costituito dal contatto ad uncino a molla con cappuccio disinseribile e dal coccodrillo del riferimento di massa; con l'impostazione 10:1 si ha un'ampiezza del segnale in ingresso all'oscilloscopio ridotta ad 1/10 rispetto a quanto applicato alla sonda. L'attenuazione 10:1 permette pertanto di estendere il range di misura di tensione che è possibile fare con l'oscilloscopio; il valore massimo di tensione applicabile alla sonda e all'ingresso dei canali dell'oscilloscopio, viene indicato dal produttore. In genere questo valore è marcato chiaramente in prossimità del BNC d'ingresso e solitamente è compreso tra 250V e 300V; fate molta attenzione perché non c'è una convenzione sul modo di indicare questa tensione, che a volte rappresenta il valore di picco, altre il valore RMS. E' molto importante che facciate sempre riferimento alle specifiche del vostro strumento e della specifica sonda che intendete utilizzare e cercate sempre di mantenere un buon margine di sicurezza, ricordandovi che in caso di errore si può facilmente causare la rottura irreparabile dello strumento ed eventualmente anche dei dispositivi connessi. L'impedenza d'ingresso dell'oscilloscopio può essere modellizzata dal parallelo di una resistenza da 1MΩ ed una capacità di valore generalmente compreso tra 10 e 30pF (il valore esatto viene indicato con precisione dal costruttore per ogni modello), che formano un polo che determina di conseguenza una risposta in frequenza del canale dell'oscilloscopio di tipo passa basso.

$$(Z_{in} = \frac{R_{in}}{1 + j2\pi f R_{in} C_{in}} \quad f_p = \frac{1}{2\pi R_{in} C_{in}})$$



Fig. 1
Sonda
passiva di alta
qualità con
attenuazione
fissa di 10:1 e
banda passante
di 500 MHz.

Per completezza si segnala che alcuni oscilloscopi, in genere di fascia medio-alta, permettono d'impostare il valore d'impedenza d'ingresso anche a 50Ω , cosa vantaggiosa se non indispensabile nelle misure ad alta frequenza e quando ci si connette con strumentazione e cavi coassiali anch'essi con impedenza di 50Ω ; come abbiamo appena visto, abbassare questo valore ha anche l'effetto di spostare più in alto la frequenza del polo, aumentando di conseguenza la banda passante, ovviamente sempre all'interno del limite massimo che permette l'elettronica dello strumento.

Per ottenere un'attenuazione pari a 10, la sonda ha al suo interno una resistenza da $9M\Omega$, che risulta essere in serie con la resistenza da $1M\Omega$ dell'oscilloscopio, con la quale realizza un partitore con il rapporto di tensione voluto, pari ad $1/10$. Se aggiungiamo un condensatore variabile in parallelo alla resistenza di attenuazione, possiamo rendere, almeno teoricamente, l'attenuazione della sonda costante ed indipendente dalla frequenza, uguagliando le due costanti di tempo di Z_1 e Z_2 (Fig.2):

$$Z_1 = \frac{R_{in}}{1 + j2\pi f R_{in}(C_{in} + C_{cavo})} \quad Z_2 = \frac{R_s}{1 + j2\pi f R_s C_{comp}}$$

$$\frac{V'_{in}}{V_{in}} = \frac{Z_1}{Z_1 + Z_2} \quad R_{in}(C_{in} + C_{cavo}) = R_s C_{comp}$$

se le due costanti di tempo risultano uguali si ha

$$\frac{V'_{in}}{V_{in}} = \frac{Z_1}{Z_1 + Z_2} \quad R_{in}(C_{in} + C_{cavo}) = R_s C_{comp}$$

Per compensare la sonda la si collega al segnale di riferimento presente su tutti gli oscilloscopi, che genera un'onda quadra tipicamente ad 1 KHz e si agisce con un piccolo cacciavite (a volte fornito insieme alla sonda) sul compensatore ceramico posto internamente, in prossimità del BNC. In

Fig. 3 possiamo vedere a sinistra il caso di sonda sovra compensata, se la capacità del compensatore è troppo alta e di conseguenza le armoniche elevate sono meno attenuate; a destra vediamo invece il caso di sonda sotto compensata perché la capacità è troppo bassa, determinando una maggiore attenuazione delle alte frequenze. Una buona compensazione si ha quando la forma d'onda risulta essere perfettamente squadrata.

Per quanto visto finora si potrebbe credere che basta utilizzare una sonda con rapporto di attenuazione 10:1 o superiore per poter estendere praticamente a piacimento il range di tensioni che si possono misurare con l'oscilloscopio, pur di rimanere all'interno delle specifiche, oppure pensare che sia possibile collegare direttamente alla rete un oscilloscopio che presenta un valore RMS massimo d'ingresso di 250 o 300V; questo non è

Fig. 2
Schematizzazione
principio di
compensazione
della sonda 10:1
dell'oscilloscopio.

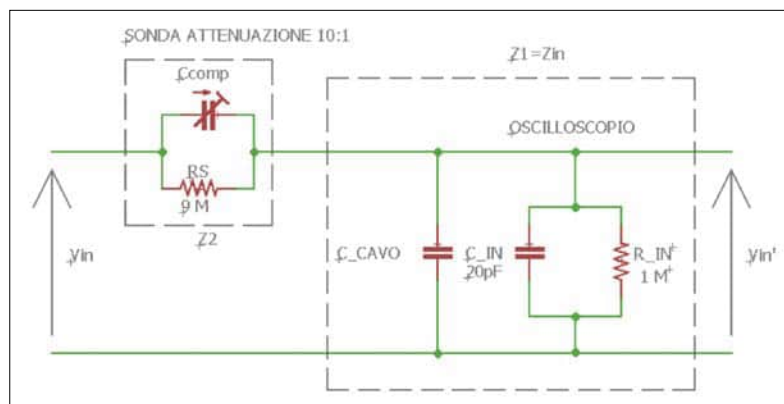
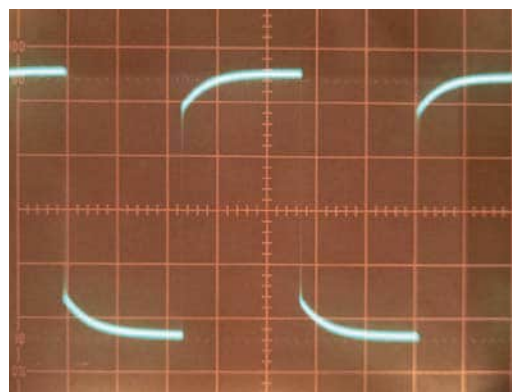
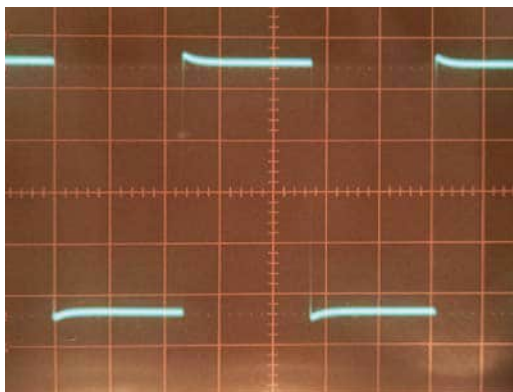


Fig. 3
A sinistra sonda
sovra compensata
e a destra sotto
compensata.



possibile perché c'è un altro fattore importante da tenere in considerazione e che soprattutto chi è alle prime armi potrebbe non sapere: il riferimento di massa di tutti i canali dell'oscilloscopio è elettricamente direttamente connesso al connettore di terra dell'oscilloscopio (nei vecchi oscilloscopi analogici non era inconsueto trovare sul pannello frontale dei comandi un morsetto a vite connesso a terra). E' un elemento da tenere in grande considerazione quando si effettuano misure con l'oscilloscopio, non soltanto quando si effettuano misure con tensioni non SELV (Safety Extra Low Voltage), eccedenti i 25V AC RMS o i 60V DC ma ogni qualvolta si collega il riferimento di massa (la clip a coccodrillo della sonda) ad un circuito si deve fare attenzione che non venga connessa a parti con un potenziale diverso da quello di terra, pena gravissimi danni anche irreparabili allo strumento stesso e al circuito sotto misura: tramite la sonda si sta collegando alla terra dell'impianto elettrico una tensione diversa da zero. Per verificare quanto appena detto, basta controllare con un multimetro in modalità continuità, che il terminale di terra dello strumento è elettricamente connesso alla parte metallica esterna dei BNC afferenti ai canali di misura dell'oscilloscopio. Quanto visto vale per quasi tutti gli oscilloscopi, tranne pochissimi rari casi di strumenti particolari dal costo non facilmente abbordabile, realizzati in modo da avere i canali fra loro isolati ed isolati dalla rete, terra compresa, utilizzati per misure specifiche. Per brevità e semplicità tralasciamo la descrizione dell'utilizzo di un trasformatore d'isolamento per risolvere il problema precedentemente evidenziato. La versatilità degli oscilloscopi può essere aumentata tramite l'utilizzo di apposite sonde che permettono di effettuare misure particolari, "estendendo" le potenzialità dello strumento e delle sonde passive; queste sonde sono di tipo attivo, avendo al loro interno componenti attivi come BJT,

MOSFET ed amplificatori operazionali e pertanto necessitano di una tensione di alimentazione per funzionare.

L'oscilloscopio è nato per visualizzare l'andamento di una grandezza elettrica corrispondente ad una tensione che varia nel tempo; tramite le sonde di corrente per oscilloscopio si può facilmente convertire una corrente in una tensione per misurare e visualizzare la forma d'onda della stessa circolante nella sezione di circuito posta all'interno della "clamp" (pinza) della sonda stessa.

SONDE DIFFERENZIALI PER OSCILLOSCOPI

Abbiamo finora visto che le comuni sonde per oscilloscopio permettono di effettuare misure di tensione con riferimento a massa (single ended), in quanto il contatto a coccodrillo che è collegato alla calza del cavo coassiale della sonda, è messo a terra tramite una connessione interna all'oscilloscopio. Per effettuare delle misure di modo differenziale con l'oscilloscopio si utilizzano delle apposite sonde: le sonde differenziali per alta tensione per oscilloscopio. Queste sonde si presentano con un corpo compatto, generalmente in plastica dovendo assicurare un ottimo isolamento elettrico con tensioni che possono anche superare i 5000V; ad un estremo ci sono due boccole, una di colore rosso e l'altra di colore nero poste come terminazione di due fili, anch'essi di colore rispettivamente rosso e nero, di ottima fattura e con grado di isolamento adeguato alla tensione che può misurare lo strumento. Dal lato opposto troviamo un cavo coassiale terminante con un connettore BNC maschio ai cui contatti è presente la forma d'onda della tensione differenziale misurata dalle due terminazioni rosse e nere che abbiamo appena visto, attenuate del fattore indicato dal selettore presente sulla corpo della sonda; questo cavetto va quindi collegato ad un canale dell'oscilloscopio. Su di un lato della sonda (a volte vicino al

cavetto coassiale d'uscita appena visto) troviamo anche un jack o una presa USB femmina che serve per l'alimentazione dello stesso (Fig. 4). È molto importante rispettare le specifiche indicate dal costruttore per la corretta alimentazione del dispositivo, come ad esempio la polarità ed il range di alimentazione applicabile nel caso di connettore a jack, pena il potenziale danneggiamento dello strumento stesso. Nel caso di alimentazione tramite connettore USB, generalmente è possibile utilizzare la presa presente sul pannello frontale (o posteriore) dell'oscilloscopio.

Alle boccole d'ingresso si possono collegare vari accessori che permettono di accedere in varie modalità alla parte sotto tensione di cui si vuole fare la misura (Fig. 5). Per questo tipo di sonde è molto importante la qualità degli accessori che si possono collegare, come puntali, coccodrilli e contatti ad uncino, questo perché dev'essere garantito un alto grado di isolamento, in genere superiore al range di misura della sonda stessa, anche dopo numerosi utilizzi degli stessi. Ricordiamo ovviamente di fare attenzione e di prendere tutte le precauzioni necessarie quando si effettuano misure su sistemi ad alta tensione.

COME SI SCELGONO

Uno dei principali parametri che caratterizzano le sonde differenziali per alta tensione per oscilloscopi è rappresentato dal massimo valore di tensione (differenziale) che può essere applicato tra i terminali di misura. Un valore molto comune è 700V ma si trovano anche facilmente sonde in grado di effettuare misure differenziali di tensioni fino a 1300V ed esistono modelli in grado di effettuare misure persino di 5600V e anche oltre. Si



Fig. 4
Sonda differenziale per alta tensione per oscilloscopio da 700V con fattore di attenuazione selezionabile tra 10:1 e 100:1 e banda passante 100 MHz.

potrebbe a questo punto pensare che può essere conveniente scegliere la sonda con il parametro di tensione massima più alto possibile ma non sempre questa è la scelta migliore. Vediamo perché. Abbiamo visto che queste sonde servono per realizzare misure differenziali di tensione tramite l'oscilloscopio e sono molto utili quando è necessario misurare tensioni su parti di circuito che non hanno un riferimento di massa equipotenziale con il riferimento di terra, cioè che presentano una tensione sul punto a cui dovremmo connettere il coccodrillo della sonda che ha un valore diverso da 0V rispetto alla terra del nostro impianto, quindi potrebbero servirci anche per effettuare misure di parti non necessariamente ad alta tensione ma semplicemente tali da presentare una differenza di potenziale minima, rispetto alla terra o comunque perché abbiamo necessità di effettuare misure differenziali su sistemi con tensioni di pochi volt.



Fig. 5
Accessori forniti insieme alla sonda differenziale per alta tensione.

In questo caso potrebbe essere utile scegliere una sonda con una tensione massima applicabile non eccessiva perché oltre al valore di tensione massima che caratterizza la sonda, questa ha due valori di attenuazione che sono sempre "scalati" rispetto alla tensione massima applicabile, in due range. Abbiamo visto che la sonda riduce l'ampiezza del segnale applicato, in modo da avere in uscita una tensione che in genere non supera i 7-10V massimi, per cui se dobbiamo misurare una tensione massima di 700V, per avere una tensione massima di 7V, dobbiamo avere un'attenuazione di 100 (infatti questo è un valore che si trova molto comunemente); per garantire una buona dinamica di tensione anche per valori di tensione più bassi (fino a 70V), si riduce l'attenuazione a 10, ottenendo sempre un valore massimo in uscita di 7V. Se invece il valore massimo che possiamo misurare è 1300V, troviamo ad esempio sonde con valore di attenuazione selezionabile di 50 e 500, in modo da avere tensioni sempre inferiori a 10V (in questo caso particolare inferiori a 3V) in entrambe le gamme di misura. Generalmente i valori di attenuazione selezionabili comunemente tramite il selettore presente sulla sonda sono: 10-100, 20-200, 50-500, 200-2000.

Quindi per decidere il valore di tensione massimo che la sonda è in grado di misurare, dobbiamo prima definire la dinamica dei segnali che intendiamo visualizzare. Normalmente una sonda con rapporto di attenuazione di 10 e 100 volte è molto utile perché permette di effettuare misure che generalmente vanno da pochi volt fino a varie centinaia, comprendendo quindi la maggior parte delle tensioni (se non tutte) che generalmente troviamo nelle applicazioni elettroniche più comuni, comprendendo anche i convertitori AC-DC (monofase), gli stadi PFC, amplificatori di potenza, etc. Abbiamo visto che è molto importante definire la dinamica del segnale che dobbiamo misurare per scegliere correttamente la tensione e in questa scelta dobbiamo necessariamente considerare un altro parametro che il costruttore ci fornisce: il rumore, generalmente indicato in millivolt RMS. Se ad esempio abbiamo una sonda con un rumore di 15mV RMS, non possiamo certo sperare di misurare segnali di ampiezza inferiore a 500mV RMS (se non superiori) in quanto questi avrebbero un'ampiezza all'ingresso del canale dell'oscilloscopio ridotta ad 1/10 (nel caso di sonda con rapporto di attenuazione 10-100), quindi di 50mV RMS, valore già paragonabile all'ampiezza del rumore. È anche vero che difficilmente avremo necessità di

misurare segnali così piccoli con la nostra sonda però è sempre meglio ricordarsi della presenza e dell'entità del rumore introdotto dalla stessa nella nostra misura.

Abbiamo finora tenuto conto del rapporto di attenuazione, del valore massimo che possiamo misurare, quindi della dinamica del segnale e del rumore che introduciamo durante la misura. Vediamo ora quali segnali possiamo analizzare, cioè consideriamo la banda passante della sonda differenziale. Come tutti gli strumenti di misura, anche la sonda differenziale ha una sua banda passante, cioè un range di frequenze in cui lo strumento può essere utilizzato. La sonda differenziale è in grado di effettuare misure che vanno dalla continua (DC) fino generalmente a 100MHz (a volte hanno frequenze di taglio inferiori, a volte superiori). Quella che viene indicata è la frequenza di taglio a -3dB, quindi il valore di frequenza per cui la tensione di uscita si riduce di 0,707 volte rispetto al valore di centro banda (supponendo la risposta in frequenza piatta, cosa che generalmente possiamo assumere). Tra le altre grandezze importanti che il costruttore fornisce vi è l'accuratezza del guadagno della sonda e di conseguenza della misura che si può effettuare con lo strumento, il CMMR (Common Mode Gain Ratio) o guadagno di modo comune, l'impedenza d'ingresso e altri ancora. Possiamo noi stessi misurare la banda passante della sonda, la sua linearità e l'accuratezza, verificando con un oscilloscopio e con un generatore di segnali delle cui calibrazioni siamo certi e verificare, scegliendo dei punti con frequenza ed ampiezza all'interno della banda passante e nel range di attenuazione selezionato, segnando come varia l'ampiezza in uscita della sonda, rispetto al segnale applicato al suo ingresso.

MITI DA SFATARE

A differenza di quanto alcuni pensano, le sonde differenziali di alta tensione per oscilloscopi non sono sonde isolate, come possiamo vedere dalla schematizzazione semplificata di **Fig. 6**: non ci sono elementi che isolano galvanicamente o in altro modo l'ingresso e l'uscita della sonda, riducibile concettualmente ad un amplificatore per strumentazione, preceduto da un attenuatore resistivo che riduce la tensione in ingresso del fattore di attenuazione massimo caratteristico della sonda (sono anche visibili alcuni condensatori di compensazione in parallelo alle resistenze d'ingresso da 1M). Nella realtà la complessità circuitale ed il numero di componenti è molto maggiore.

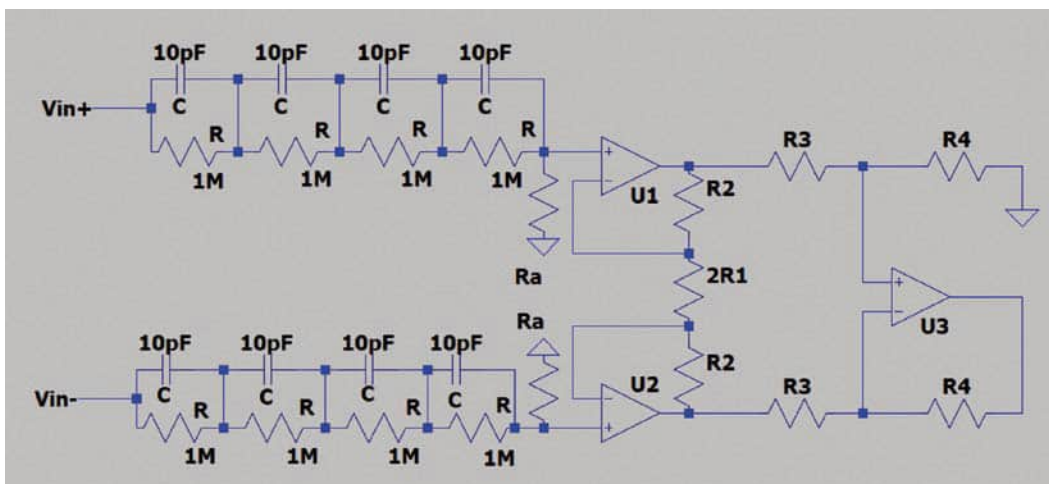


Fig. 6
Schema
semplificato della
sonda differenziale
con l'attenuatore
d'ingresso di
alta tensione e i
condensatori di
compensazione.

ESEMPIO DI MISURE

Ci è chiaro, da quanto visto finora, che non possiamo collegare direttamente la sonda passiva dell'oscilloscopio, anche se attenuata, direttamente alla rete elettrica. A meno di non avere uno strumento con i canali adeguatamente isolati (strumento molto costoso e poco comune).

Con la sonda differenziale invece è possibile farlo, dobbiamo solo impostare il corretto valore di attenuazione, facendo comunque attenzione al rispetto delle norme di sicurezza elettrica e collegare i terminali della stessa alla rete elettrica, servendoci preferibilmente di un accessorio che garantisca un contatto stabile come ad esempio i grossi coccodrilli oppure le pinzette ad uncino. In **Fig. 7** è visibile la forma d'onda della tensione di rete, ottenuta utilizzando una sonda differenziale da 700V, impostata per avere un'attenuazione di 100:1, collegata al primario di un piccolo trasformatore. La stessa figura mostra anche il confronto della tensione di rete a 230V RMS (325V di picco) con una sinusoide spettralmente sufficientemente pura.

Dalla figura si può notare la leggera distorsione della forma d'onda di rete, che mostra come lo spettro della tensione disponibile alle nostre prese ha, oltre alla fondamentale, delle armoniche a frequenze superiori ai 50Hz, che ne causano la distorsione visibile.

CONCLUSIONI

Non è possibile collegare il riferimento di massa dell'oscilloscopio in qualunque parte di un circuito elettrico perché questo è collegato elettricamente alla terra e potremmo causare gravi guasti alle attrezzature. Per misurare in sicurezza tensioni di rete o tra due punti che possono essere a qualsiasi potenziale rispetto a quello di massa (nel caso dell'oscilloscopio coincidente con la terra) si utilizza la sonda differenziale che va scelta coerentemente con il tipo di misure che si devono realizzare (valore massimo, dinamica, rumore, banda passante). Bisogna comunque ricordare che la sonda differenziale non isola l'oscilloscopio dal circuito al suo ingresso ma semplicemente effettua un'attenuazione precisa della tensione ai suoi capi. ■

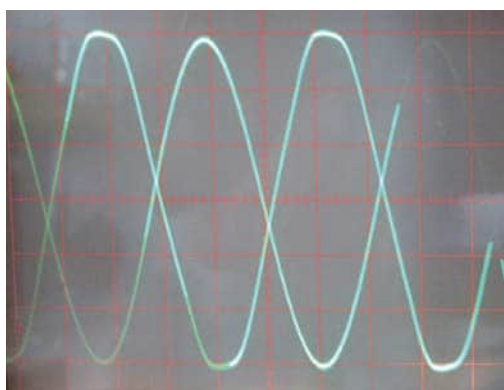
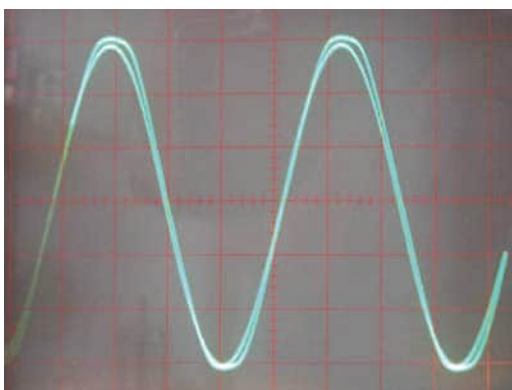


Fig. 7
Confronto fra
una sinusoide
spettralmente
sufficientemente
pura e la
sinusoide della
tensione di rete.

Per lo smart working, per chattare con amici e parenti comodamente da casa!

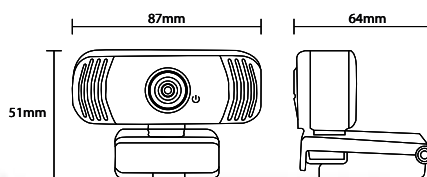
Prezzo IVA inclusa.



€24,00
Cod. FR795

WEBCAM FULL HD 1080P

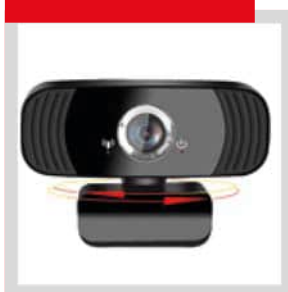
Webcam Full HD 1080P particolarmente indicata per seguire lezioni online, effettuare videoconferenze, videochat, videosorveglianza, registrazioni video, ecc. Dispone di lenti prive di distorsione, microfono omnidirezionale con riduzione del rumore, regolazione manuale della messa a fuoco, angolo di ripresa: 60° (verticale) – 120° (orizzontale), rotazione: 360°, LED di stato per alimentazione e attivazione webcam. Completamente Plug & Play (non richiede l'installazione di nessun driver), può funzionare con Windows 10/8/7/ Vista, Mac OS X, Linux, ChromeOS, Ubuntu e Android 4.0 o versioni successive. Basta collegarla alla porta USB del PC, attendere che il sistema operativo la rilevi e poi utilizzarla con Skype, Zoom, Facebook, ecc. Grazie alla sua base multifunzionale è possibile posizionarla e utilizzarla su PC fissi, PC portatili, smart TV, appoggiarla sulla scrivania oppure montarla su treppiede per macchine fotografiche.



- ✓ **Processore:** 3518EV300
- ✓ **Sensore immagine:** CMOS a colori
- ✓ **Risoluzione:** 1920 x 1080
- ✓ **Video stream:**
 - H.264: 1920 x 1080 / 1920 x 720 / 640 x 360
 - MJPEG: 1920 x 1080 / 1920 x 720 / 640 x 360
 - YUV: 640 x 360
- ✓ **Video frame rate (fps):** 30 fps (1280 x 720 / 1920 x 1080)
- ✓ **Uscita Video:** USB Video streaming
- ✓ **Controllo Immagine:**
 - messa a fuoco manuale
 - bilanciamento del bianco automatico
 - controllo automatico dell'esposizione (AEC, Automatic Exposure Control)
- ✓ **Otturatore elettronico:** automatico
- ✓ **Microfono:** omnidirezionale
- ✓ **Rapporto segnale/rumore:** > 50dB
- ✓ **Angolo di ripresa:** 60° (verticale) 120° (orizzontale)
- ✓ **Rotazione:** 360°
- ✓ **Sistemi operativi supportati:** Windows 10/8/7/Vista, Mac OS X, Linux, ChromeOS, Ubuntu e Android 4.0 o versioni successive
- ✓ **Alimentazione:** 5VDC / 160mA (tramite porta USB)
- ✓ **Temperatura di lavoro:** da -40°C a +70°C
- ✓ **Materiale:** ABS
- ✓ **Dimensioni (mm):**
 - Webcam: 87 x 64 x 51mm
 - lunghezza del cavo USB: 1,5m
- ✓ **Peso:** 180g



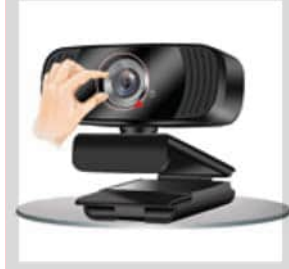
Ruotabile di 360°



Vari metodi di posizionamento



Supporta la messa a fuoco manuale



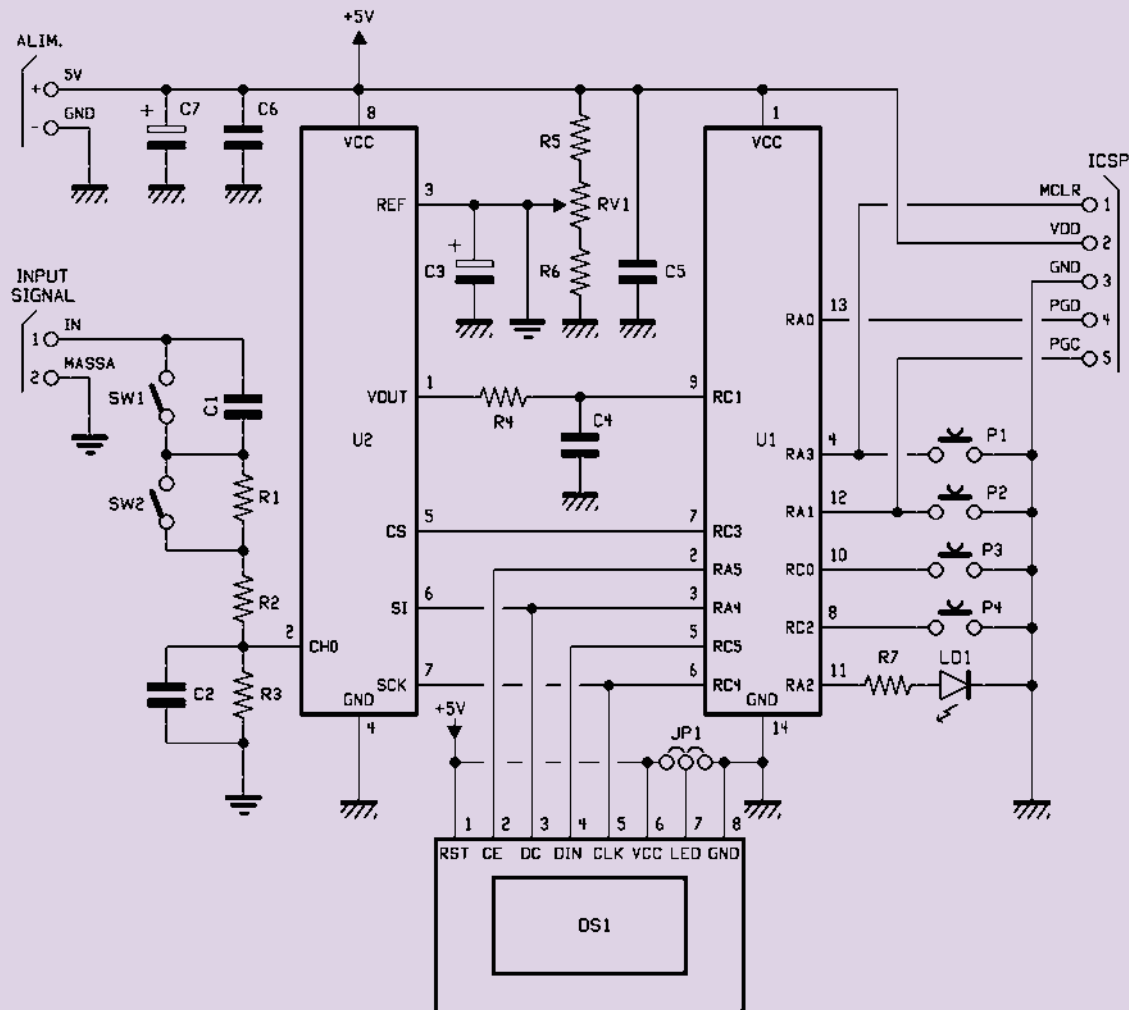
MICRO OSCILLOSCOPIO



di **SILVANO BREGGION**

Indispensabile nel nostro laboratorio di elettronica, ha le funzioni basilari di uno strumento economico e la base dei tempi da 500 kps per misure in BF.

Non è la prima volta che proponiamo il progetto di un piccolo oscilloscopio per eseguire misure essenziali senza spendere i soldi richiesti dall'acquisto di uno strumento dalle caratteristiche professionali; però questa volta lo facciamo con qualcosa di semplice e facile da realizzare impiegando, come visualizzatore, il display di un telefono cellulare molto diffuso in passato e pertanto facilmente reperibile sul web. Per quanto non siano spinte, le prestazioni di questo oscilloscopio sono sicuramente interessanti e potrebbero essere molto utili nel laboratorio di un hobbista. Il maggiore limite dello strumento è il numero dei campionamenti per secondo, che arriva a 500 kps e permette quindi la visualizzazione di segnali a frequenza inferiore a 50 kHz; questo perché decifrare la forma d'onda a 50 kHz con 10 campionamenti al secondo diventa difficile; se si tratta di un'onda quadra il segnale può



essere visualizzato in maniera decente, mentre se il segnale è una sinusoide è difficile distinguere quest'onda da una triangolare. Lo strumento nasce principalmente per trovare impiego tra le mani dell'hobbista; peraltro la presenza di marker si dimostra molto utile sia per misurare il periodo del segnale campionato, sia per misurare la relativa ampiezza. Un esempio pratico di utilizzo è la verifica del segnale di clock di un microcontrollore PIC, dove per impostare la frequenza dell'oscillatore interno è necessario parametrizzare in maniera precisa un registro che controlla l'hardware del micro; la cosa può sembrare banale ma a noi è successo più volte di perdere tempo nel cercare di capire perché la velocità di certe funzioni è sbagliata. L'esperienza ci consiglia di partire sempre con un loop che esegue il toggling su un piedino del PIC, seguito da un ritardo di 200 millisecondi.

Se misuriamo in maniera corretta tale ritardo con l'oscilloscopio, possiamo cancellare il semplice loop e proseguire in maniera spedita alla stesura del programma. Nel PIC16/PIC18 il modulo hardware dell'oscillatore viene programmato da uno o due registri; nel caso del PIC32, dove i registri interessati sono molti e c'è da impazzire nel parametrizzarli in maniera corretta, l'oscilloscopio diventa indispensabile.

Altra caratteristica dello strumento proposto è il livello di soglia del segnale da analizzare, chiamato *trigger*, che ci permette di vedere cosa succede dopo che il segnale applicato all'ingresso supera un certo valore. Purtroppo i campionamenti sono limitati a 80 e ne spiegheremo più avanti il perché.

SCHEMA ELETTRICO

La semplicità è l'elemento distintivo dello schema elettrico. Di primo acchito può fare pensare che

il progetto abbia delle caratteristiche non idonee al normale utilizzo, ma in realtà una buona parte delle funzioni richieste viene svolta dai moduli hardware interni al PIC. Oltre all'ADC (Analog Digital Converter) necessario per rilevare i vari campionamenti, il timer TMR6 per temporizzare i campionamenti lunghi, sfruttiamo il convertitore DAC (Digital Analog Converter) e un comparatore completo di rilevatore di soglia, componenti necessari per la gestione del trigger. Se il PIC ne fosse stato sprovvisto, avremmo dovuto complicare il circuito dotando lo stesso aggiungendoli. Descriviamo lo schema elettrico seguendo il segnale da analizzare, a partire dall'ingresso: il condensatore C1 (bypassabile con lo switch SW1) è necessario per bloccare la componente continua quando il segnale da analizzare è alternato; le resistenze R2 e R3 formano un partitore resistivo fisso che divide il segnale per 10. La resistenza R1 può essere cortocircuitata dallo switch SW2; se SW2 viene aperto, il partitore resistivo divide per 20. Il valore massimo del segnale in ingresso è di 50V con SW2 chiuso. Con SW2 aperto l'ampiezza massima misurabile è di 100V.

Il segnale passato da questo primo stadio viene portato all'ingresso di U2, un PGA (Programmable Gain Amplifier) che permette di ridurre di molto il numero di componenti necessari. A questo punto dobbiamo fare una precisazione: la soluzione adottata non è canonica, perché il partitore resistivo fisso posto all'ingresso avrebbe dovuto essere sostituito da un partitore multiplo in qualche modo controllato dal PIC. Anche l'amplificatore che segue, necessario per adattare l'impedenza e non caricare il partitore resistivo, avrebbe dovuto essere dotato di un guadagno in qualche modo

variabile, sempre gestito dal PIC.

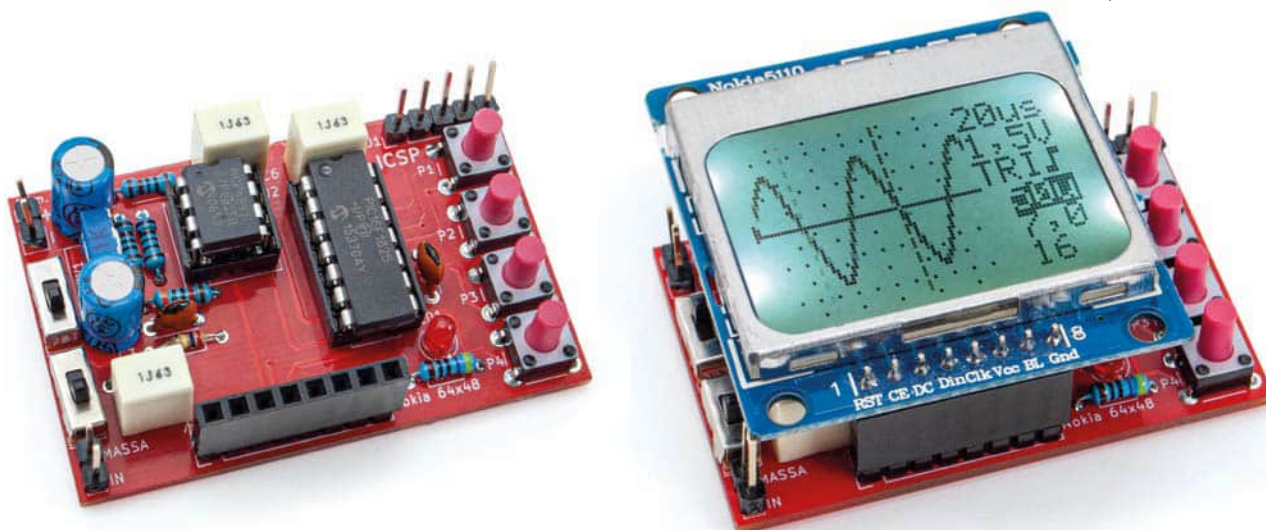
Il tutto avrebbe richiesto un numero non indifferente di componenti in più e un microcontrollore ben fornito di pin necessari alla gestione del partitore/amplificatore. Il PGA in questione, MCP6591 di casa Microchip, ha il guadagno programmabile in 8 step. Gli 8 step sono: +1, +2, +4, +5, +8, +10, +16, +32. Con il guadagno +1 il PGA fornisce in uscita la stessa ampiezza presente nell'ingresso non invertente, cioè divisa per 10 dal partitore resistivo. Con il secondo step il segnale viene posto in uscita diviso per 5 e così via fino allo step +10 che fornisce la stessa ampiezza del segnale posto ai morsetti in ingresso. Gli ultimi 2 step amplificano il segnale rispettivamente 1,6 volte e 3,2 volte. Il PGA viene gestito dal PIC attraverso un'interfaccia seriale sincrona formata da SCK (CLOCK), SI (Serial Data Input) e CE (Chip Select). I due pin SCK e SI sono in comune con il display, quindi il PGA viene gestito integralmente da unico pin.

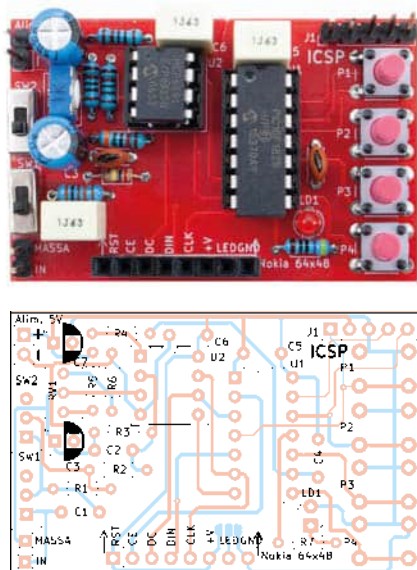
Il trimmer RV1 ha il compito di fornire il riferimento al pin 3 del PGA e la massa fittizia ai morsetti di ingresso. Anche questa soluzione, nonostante l'efficacia, andrebbe sostituita con la doppia alimentazione per l'amplificatore, per visualizzare il segnale alternato. Con l'alimentazione singola, è possibile visualizzare solo la parte positiva del segnale ed escludere quella negativa. Il trimmer RV1 e componenti associati, risolvono il problema creando la massa fittizia regolata a 2,5V. Da notare che la massa realizzata da RV1 è diversa dalla massa generale (GND), motivo per cui è necessario alimentare lo strumento con una alimentazione dedicata. Ipotizziamo la necessità di testare un segnale generato da un altro PIC e di

CARATTERISTICHE TECNICHE

- ➔ **Base dei tempi:**
20us, 40us, 70us, 500us, 1ms, 5ms, 10ms, 50ms, 100ms, 500ms, 1s, 10s
- ➔ **Ampiezza verticale:**
240mV, 460mV, 750mV, 1V, 1.5V, 1.8V, 3.7V, 6.9V
- ➔ **Selzione tipo di avvio campionamento:** RUN, TRIGGER, ONCE SHOT
- ➔ **Su "Trigger" e "Once shot" impostazione del punto di partenza dei campioni**
- ➔ **Selezione fronte in salita/fronte in discesa**
- ➔ **Misura dei tempi con i Marker verticali**
- ➔ **Misura delle ampiezze con i Marker orizzontali**
- ➔ **Possibile visualizzazione di tutti i campioni a schermo intero**

↓
La scheda dell'oscilloscopio con e senza display.





Elenco Componenti:

R1: 300 kohm	DIS.1: Display Nokia 5110
R2: 270 kohm	SW1-SW2: Micro switch levetta
R3: 30 kohm	P1-P2-P3-P4: Pulsante 6mm
R4-R5-R6: 1 kohm	
R7: 470 ohm	
RV1: trimmer 1 kohm	Varie:
C1-C5-C6: 0,1 uF poliestere	- zoccolo 4+4
C2-C4: 22pF ceramico	- zoccolo 7+7
C3-C7: 100uF 25V elettrolitico	- strip femmina 8 poli
LD1: LED rosso 3mm	- strip maschio 5 poli
U1: PIC16F1825	- strip maschio 2 poli
U2: MCP6S91	- stampato s1564 (54x35)

alimentare il tutto con un'unica alimentazione, la visualizzazione non sarà corretta. Non dobbiamo, quindi, confondere la massa fittizia (GNDREF) con la massa generale (GND).

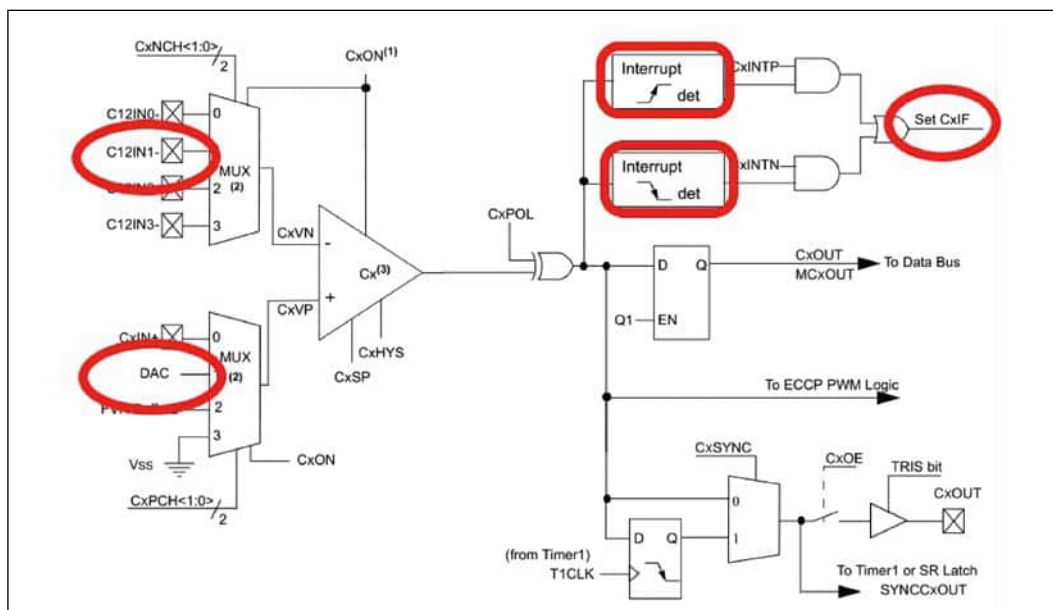
Il segnale in uscita dal PGA viene collegato al RC1 del PIC16F1825, a sua volta internamente collegato all'ADC e, configurando opportuni registri (CM1CON0 e CM1CON1), anche all'ingresso invertente del comparatore CM1.

I due comparatori presenti nel PIC possono essere usati come componenti indipendenti, con la possibilità di collegare fisicamente gli ingressi e le uscite ad alcuni pin del PIC.

Nella nostra applicazione abbiamo impostato i registri del comparatore in modo da collegare il DAC all'ingresso non invertente. Il DAC è in grado di fornire 32 livelli da 0 a 5V. Sull'uscita del comparatore è presente il detector per decidere su quale fronte vogliamo che parta il campionamento. Dalla **Fig.1** è possibile prendere visione del principio di funzionamento molto semplificato del rilevatore di trigger.

Il display utilizzato è un componente facilmente reperibile a basso prezzo e molto utile in molteplici occasioni. Si tratta del modello usato dal vecchio Nokia 5110 e simili modelli; la sua definizione è relativamente elevata, 84x48 pixel, tanto che può sostituire egregiamente il display 16x2 standard in tutte le situazioni dove sorge la necessità di visualizzare delle semplici scritte o valori vari.

Fig. 1
Schematizzazione del rilevatore di trigger implementato dal PIC.



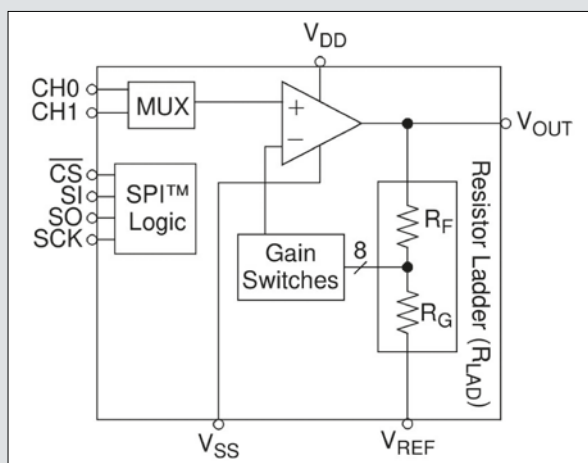
I piedini necessari per gestire il GLCD Nokia sono solo 6, più i due di alimentazione. In genere il RESET viene inizializzato solo al momento dell'alimentazione del circuito; in tal caso può essere posto direttamente al +Vcc per risparmiare un pin del microcontrollore. Quanto alla retroilluminazione a LED, non tutti i modelli utilizzano la stessa configurazione: in alcuni casi il relativo contatto

va collegato al +Vcc, mentre in altri tale piedino va collegato a GND. Nella nostra applicazione abbiamo collegato il RESET direttamente al +Vcc e abbiamo previsto il ponticello JP1 (lo vedete in Fig. 2) per selezionare il polo corretto per alimentare la retroilluminazione. In totale, sono quattro i pin collegati al micro. Come anticipato, due piedini sono in comune con il PGA.

PGA MCP6S91

L'integrato MCP6S91 di casa Microchip, è un amplificatore operazionale con guadagno variabile in 8 step, gestibile dal microcontrollore in seriale sincrona (SPI) a tre fili.

È un chip che presenta delle caratteristiche interessanti come la larghezza di banda che arriva a 18 MHz, rail to rail e alimentazione singola. In commercio si trova un modello con due ingressi multiplexabili (MCP6S92) e uno che offre la possibilità di collegare più PGA in serie sempre con tre fili (MCP6S93).



Il PGA è interamente gestito da tre registri, dei quali vediamo di seguito i dettagli.

INSTRUCTION REGISTER							
W-0	W-0	W-0	U-x	U-x	U-x	U-x	W-0
M2	M1	M0	—	—	—	—	A0
bit 7							bit 0

Si tratta del primo registro che abilita la scrittura dei successivi registri. In particolare A0=1 per indirizzare l'ingresso, A0=0 per selezionare il guadagno. Nel nostro caso usando un PGA con singolo ingresso, A0 va sempre posto a zero.

M2, M1 e M0 richiedono un codice per abilitare la scrittura (010) o per risparmiare energia ponendo in Shutdown il PGA(001).

GAIN REGISTER							
U-x	U-x	U-x	U-x	U-x	W-0	W-0	W-0
—	—	—	—	—	G2	G1	G0
bit 7							bit 0

Selezione di otto step di guadagno:

000 = Gain of +1
 001 = Gain of +2
 010 = Gain of +4
 011 = Gain of +5
 100 = Gain of +8
 101 = Gain of +10
 110 = Gain of +16
 111 = Gain of +32

CHANNEL REGISTER							
U-x	U-x	U-x	U-x	U-x	U-x	U-x	W-0
—	—	—	—	—	—	—	C0
bit 7							bit 0

Il C0 seleziona il canale (ingresso) desiderato, C0=0: canale 0, C0=1: canale 1. In uso per MCP6S92 e MCP6S93.

Nel pacchetto trovate la libreria per gestire il PGA nel file PGA_MCP6S9X.h. Non usa moduli hardware, quindi può essere usufruito da tutti i microcontroller. La funzione principale è molto semplice void PGA_Guadagno (unsigned short guadagno); invia la richiesta di scrittura seguita dal Byte di indirizzamento del guadagno, come di seguito:

```
void PGA_Guadagno (unsigned short guadagno) {
    if (guadagno > 0x7) guadagno = 0x07;
    ENEBLE_PGA1 = 0;           // Enable CS del MCP6S9x
    Tx_PGA (0b01000000);      // PGA ON, select gain
    Tx_PGA (guadagno);         // Gain
    ENEBLE_PGA1 = 1;           // Disable CS del MCP6S9x
}
```

Concludiamo con i quattro pulsanti necessari all'utilizzo pratico dello strumento e il LED che ha un compito superficiale, si illumina brevemente quando il comparatore rileva il passaggio del segnale da misurare sul livello imposto dal trigger. Il circuito è alimentato con 5V stabilizzati.

MONTAGGIO

Ci sono alcuni componenti che possono essere esclusi dal montaggio. R1 può essere sostituito da un ponticello e lo switch SW2 omissso dal circuito. Il LED può essere utile ma non influisce sul funzionamento. Consigliamo caldamente l'uso di zoccoli per i due integrati.

Stesso discorso per l'installazione del display, un connettore è necessario per poterlo sfruttare in altri progetti.

Il connettore J1 merita un discorso a parte: se desideriamo modificare il firmware per testare alcune idee che ci passano per la testa, è utile montare il connettore per tenere sempre collegato il programmatore, altrimenti potete programmare il PIC direttamente sul programmatore prima di inserirlo nello zoccolo ed escludere il connettore dal montaggio.

Ricordatevi di collegare la retroilluminazione del display ponticellando con una goccia di stagno le piazzole siglate JP1 sul piano di montaggio.

FIRMWARE

A livello firmware abbiamo implementato quattro tipi di campionamento:

- il primo, veloce, a 20µs;
- il secondo, a 40µs, basato su temporizzatore fisso;
- il terzo con temporizzatore variabile da 70µs fino a 10ms;
- l'ultimo, variabile da 50ms a 10s, che abbiamo chiamato "real time" perché il tempo tra un campionamento e l'altro è più grande del tempo impiegato a disegnare un colonna sul display.

Con quest'ultimo, lo strumento campiona e traccia sullo schermo, mentre negli altri casi il display viene aggiornato integralmente dopo avere completato gli ottanta campioni.

Per farvi capire il perché sono stati scelti 80 campioni, dobbiamo descrivere l'organizzazione della memoria RAM. Nei PIC12/16 la memoria RAM è organizzata in più pagine chiamate BANK. Ogni banco presenta 128 locazioni di memoria divisa in registri di uso generale (Core Registers), registri di gestione dei moduli hardware SFR (Special Function Registers), una area di 80 Byte massimi di memoria RAM (General Purpose) e un'area di 16 Byte di memoria Common RAM, ovvero la troviamo in tutte i banchi e dobbiamo precisare che l'ambiente di sviluppo MikroC le usa di routine per gestire la compilazione da Ansi C a linguaggio macchina, sono nominate da R0 a R15. Non possiamo usarle per non incorrere in arresti indesiderati del PIC. Nei PIC12/16 non esiste un Program Counter da 16 bit; i puntatori sono due: il primo punta alle 128 locazioni di memoria di ogni banco, mentre il secondo punta al banco. Il comando che scrive sul secondo Byte (BSR) è MOVLB XX, dove XX è il numero del banco in cui si trova il registro o la locazione di memoria RAM, da 0 a 31.

Si tratta comunque di un impulso di clock che diventano due perché deve essere richiamato il banco in cui si trovano i registri che gestiscono il ADC con il compito di caricare nell'accumulatore il valore immagazzinato in ADRESH, il secondo richiama il banco in cui si trova la memoria RAM e deposita il valore contenuto nell'accumulatore. I due impulsi di clock menzionati ritardano l'acquisizione del campionamento. A questo punto apriamo il datasheet del PIC16F1825 al paragrafo "3.0 MEMORY ORGANIZATION" e cerchiamo il banco in cui trovare i registri del ADC: si trovano nel BANK 1. Lo stesso banco dispone di 80 Byte di memoria RAM e decidiamo di usare questi 80 Byte per risparmiare sulla velocità di esecuzione del campionamento. Per farlo è sufficiente dichiarare il vettore con l'indirizzo assoluto desiderato, nel nostro caso:

```
char sample_v1[80]absolute 0x0A0;
```

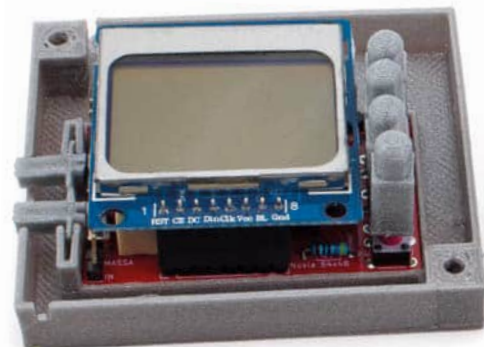
Il vettore va diviso in singole variabili:

```
char v1 absolute 0x0A0, v2 absolute 0x0A1, ecc.
```

Il tutto si trova al banco 1. Se volessimo usarne di più è possibile ma richiede diverse volte il coman-

Fig. 2
Il ponticello JP1 è collocato lato saldature e si chiude con una goccia di stagno.





←
L'oscilloscopio
nel suo
contenitore
stampato in 3D.

do che identifica il banco, quindi più tempo per campionare, rendendo impossibile raggiungere la velocità sperata di 500 ksp/s.

TARATURA E UTILIZZO

La calibrazione dello strumento passa dalla taratura del trimmer RV1, la quale è molto semplice: per operarla bisogna innanzitutto mettere in cortocircuito i morsetti di ingresso del segnale e ruotare il cursore del trimmer in modo da fare coincidere la riga del segnale mancante con la mezzeria dello schermo del display. I quattro pulsanti hanno una funzione intuitiva: partendo dall'alto troviamo il pulsante UP, segue DOWN, quindi SET e per finire END. Il pulsante SET seleziona il campo desiderato partendo dall'alto verso il basso in un loop continuo. Il campo puntato da SET visualizza il valore in contrasto.

Iniziamo dal primo campo indicato da SET, la base dei tempi. I pulsanti UP e DOWN permettono di selezionare uno dei dodici tempi a disposizione. Il secondo campo seleziona l'ampiezza sempre usando i pulsanti UP e DOWN. In questo campo possiamo scegliere il fronte del trigger con il pulsante END. Ovviamente il fronte funziona solo se abbiamo selezionato il modo TRIGGER o ONCE. Il terzo campo seleziona il modo di campionare con il pulsante END. Possiamo scegliere tra tre modi: RUN (RUN), TRIGGER (TRI) e ONCE (ONC) in loop continuo. Con TRIGGER possiamo visualizzare dei segnali ben fermi sul display con campionamenti di segnali ripetitivi, mentre con ONCE viene visualizzato l'ultima serie campionata. Nel modo RUN la visualizzazione risulta disordinata perché campiona in modo continuo senza tenere conto del livello del trigger. Può sembrare inutile ma a volte si ha la necessità di rilevare dei segnali che non passano per il livello del trigger, in questo caso diventa indispensabile selezionare il modo

RUN per regolare il livello del trigger usando i pulsanti UP e DOWN. Nell'ultimo campo possiamo notare il livello del trigger in un valore indicativo e non significativo.

I due campi successivi riguardano i marker di orientamento verticali e orizzontali: i primi misurano il tempo tra i due marker mentre quelli orizzontali misurano l'ampiezza. In questi campi sono abilitati i pulsanti UP e DOWN per movimentare i marker verticali da sinistra verso destra e viceversa, mentre i marker orizzontali dall'alto in basso e viceversa. Il pulsante END seleziona il marker sinistro/alto, destro/basso, oppure pulisce il display dai marker anche in questo caso in loop continuo. Siamo arrivati all'ultimo campo che ci permette di visualizzare tutti gli ottanta campioni, escludendo le informazioni sulla destra del GLCD premendo il pulsante END.

Quando scegliamo questa funzione, resta attivo solo il pulsante END.

Qualcuno lo definisce "gadget", non è decisamente uno strumento professionale, ma nei suoi limiti questo oscilloscopio non mancherà di soddisfare le vostre aspettative. □



Cosa occorre?

I componenti utilizzati in questo progetto sono di uso comune e facilmente reperibili. Il PCB può essere realizzato utilizzando il nuovo servizio di stampa circuiti stampati (cod. PCBPRODUCTION) direttamente dal sito www.futurashop.it.

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



HDMI
↓
VGA

CONVERTITORE DA HDMI A VGA 1080P PLUG AND PLAY

Permette di convertire un segnale digitale HDMI (compatibile con HDMI1.3) in un segnale VGA (SUB-D) ad alta definizione con audio in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso VGA.

€ 11,50

Cod. **BOXHDMITOVGA**

Convertitori di segnale

Permettono di convertire un segnale digitale HDMI in VGA, AV in HDMI, VGA in HDMI, HDMI in AV.

CONVERTITORE DA AV A HDMI PLUG AND PLAY



AV
↓
HDMI

€ 14,00

Cod. **BOXAVTOHDMI**

Permette di convertire un segnale audio/video CVBS (PAL e NTSC) in ingresso in un segnale HDMI in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita CVBS possono connettersi a schermi con ingresso HDMI. L'uscita HDMI supporta una risoluzione fino a 1080P 60Hz.



VGA
↓
HDMI

CONVERTITORE DA VGA A HDMI PLUG AND PLAY

Permette di convertire un segnale audio e video VGA (SUB-D) in un segnale HDMI (tipo A) ad alta definizione con audio. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita VGA possono connettersi a schermi HD di grandi dimensioni oppure a proiettori HD con ingresso HDMI. L'ingresso VGA supporta video fino a 1920x1080 a 60 Hz, l'uscita HDMI può supportare video fino a 1080P.

€ 11,50

Cod. **BOXVGATOHDMI**

CONVERTITORE DA HDMI AD AV PLUG AND PLAY

Permette di convertire un segnale HDMI in ingresso in un segnale video composito CVBS (PAL e NTSC) in uscita. Con questo convertitore, il computer, il lettore video portatile, le console di gioco e alcuni altri dispositivi video con uscita HDMI possono connettersi a schermi con ingresso CVBS. L'ingresso HDMI supporta una risoluzione fino a 1920x1080 a 60 Hz.



HDMI
↓
AV

€ 11,00

Cod. **BOXHDMITOAV**

TOTEM ROBO BOARD X4

Impariamo a sviluppare applicazioni e a programmare con la RoboBoard X4 di Totem, la piattaforma per la robotica educativa dell'ecosistema Totem.

di **DAVIDE SCULLINO**

M

olti progetti di robotica sono accomunati da determinati requisiti: essere in grado di controllare i motori e servocomandi, disporre di interfacce di controllo e programmazione e poter funzionare autonomamente alimentati da batterie. Tutto ciò richiede un hardware adeguato ed un controllore programmabile versatile, come può essere quello di Totem (totemaker.net) che offre una soluzione all-in-one con driver e connessione wi-fi e Bluetooth, chiamata RoboBoard.



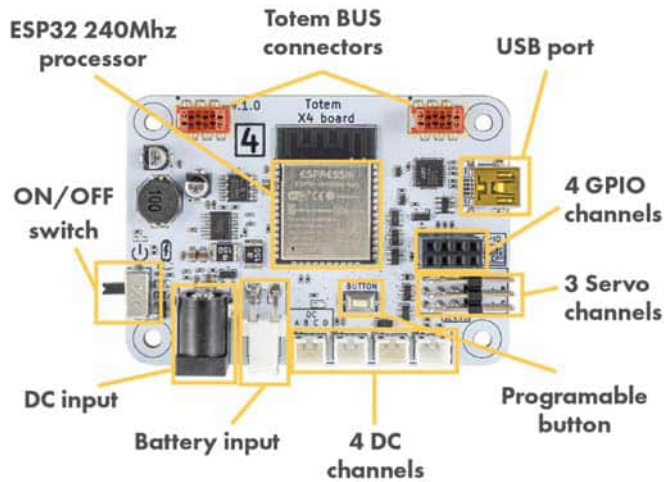


Fig. 1
La scheda
Totem
RoboBoard.

Qui descriveremo le proprietà e il principio di funzionamento, quindi passeremo alla pratica proponendo alcuni progetti di esempio, che coprono diversi casi d'uso. Totem RoboBoard (la vedete dettagliata nella **Fig. 1**) implementa le funzionalità di base che vengono utilizzate più spesso nei pro-

getti di robotica ed elettronica, che nello specifico sono:

- un alimentatore a doppio ingresso; l'adattatore di alimentazione può essere utilizzato come fonte di alimentazione primaria, così come una batteria opzionale può essere utilizzata come fonte di alimentazione di backup o principale;
- driver per motori CC, in numero di quattro canali da 1A con protezioni integrate; i driver possono essere collegati a ponte per il controllo di motori che assorbono fino a 2A;
- collegamenti servomotore (3 canali che possono fornire fino a 1A ciascuno);
- 4 connessioni ingresso/uscita per uso generico;
- LED RGB controllabile dall'utente;
- ampia connettività, comprendente Bluetooth, interfacce WiFi e una coppia di connettori TotemBUS per estensioni.

Tutta la connettività I/O è protetta contro sovraccarico o collegamenti errati per evitare danni alla scheda. Inoltre, utilizzando l'interfaccia servo standard, è possibile gestire controller RC e driver motore di terze parti senza alcuna difficoltà.

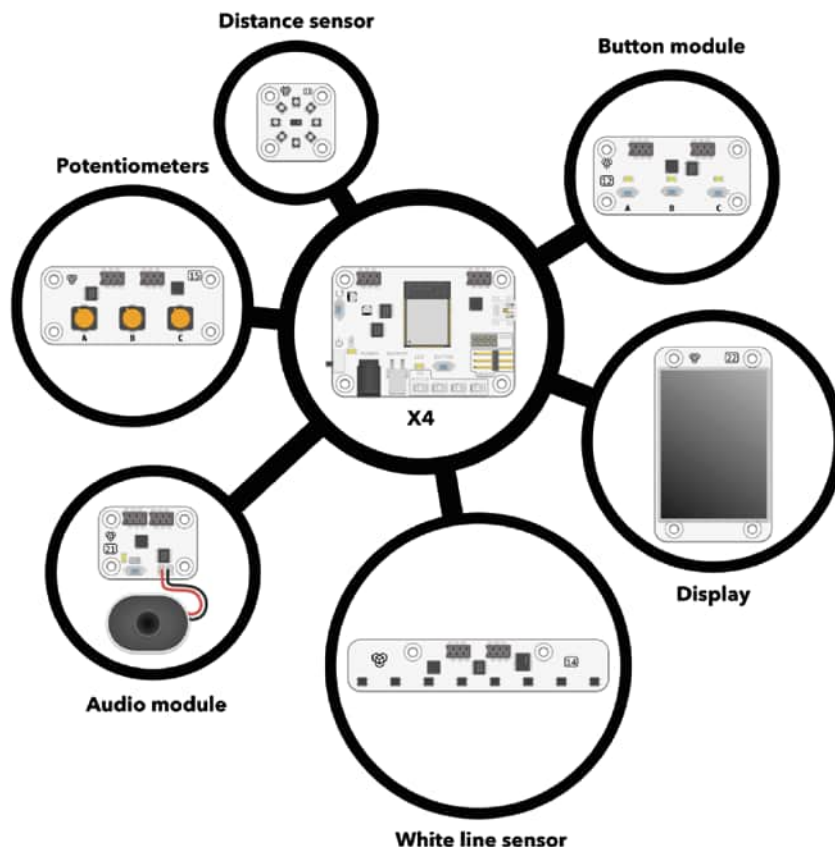


Fig. 2
Schede di
espansione
Totem.

ESPANSIONI

Alla scheda Totem RoboBoard, che è il cervello di ogni applicazione, possono essere associati moduli di espansione, ognuno dei quali svolge una funzione periferica specifica per la robotica (**Fig. 2**). Collegando moduli aggiuntivi tramite il canale TotemBUS integrato si estendono le capacità del sistema. I moduli attualmente disponibili includono sensori di distanza, riflettività, dispositivi di input come potenziometri, pulsanti e moduli di interfaccia come audio e display. Ogni modulo può essere immediatamente utilizzato dopo essere stato collegato; nessuna configurazione o programmazione è necessaria per utilizzare i moduli d'interfaccia. L'interfacciamento con i moduli di espansione avviene tramite un bus proprietario chiamato TotemBus

TOTEMBUS

I moduli collegati alla RoboBoard formano una rete a margherita, poiché ogni modulo ha almeno due connettori TotemBUS che consentono ulteriori estensioni. L'alimentazione viene trasmessa sullo stesso cavo a 6 poli, richiedendo un solo filo indipendente dal numero di moduli collegati. TotemBUS utilizza l'interfaccia CANBUS per una trasmissione dati veloce, affidabile e a lunga distanza. Inoltre nella comunicazione viene utilizzato il sistema di messaggistica in stile MQTT, per lo scambio di messaggi tra i moduli. Ogni modulo può trasmettere dati ("pubblicare" un messaggio) in qualsiasi momento e chiunque sia interessato a riceverne dei contenuti può richiedere l'inoltro di tali messaggi.

Ciò rende il sistema veloce, in quanto non è necessario eseguire il polling di nuovi dati, poiché vengono trasmessi immediatamente.

CONTROLLO DI ROBOT

Il sistema Totem RoboBoard supporta due modalità di controllo: la prima attraverso una specifica app per smartphone e la seconda tramite programmazione in ambiente Arduino.

Entrambi i metodi possono essere utilizzati allo stesso tempo e si complementano a vicenda.

APPLICAZIONE PER SMARTPHONE

L'app fornisce il punto di partenza più semplice per interagire con RoboBoard e agisce come un telecomando personalizzabile per esso. Permette di seguire le istruzioni fornite, oppure creare le proprie interfacce di controllo trascinando e rilasciando vari widget, come pulsanti, joystick e slider.



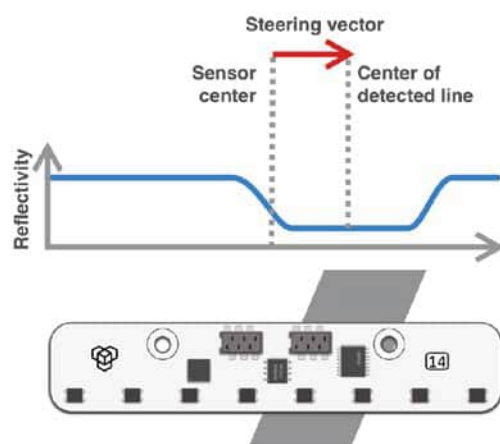
Fig. 3
Applicazione totem per smartphone.

Quindi è alla portata di tutti perché per utilizzarla non è richiesta alcuna codifica o conoscenza avanzata; l'app (la cui schermata è proposta nella **Fig. 3**) consente di creare qualcosa di unico e personale. L'app è disponibile per dispositivi sia Android che iOS e può essere scaricata cercando "TotemMaker" nei rispettivi app store.

PROGRAMMAZIONE ARDUINO

Per i casi più avanzati, in cui è necessaria la programmazione, RoboBoard può funzionare anche come dispositivo compatibile con Arduino. Per questo viene fornita una libreria speciale che consente agli utenti di utilizzare le funzionalità integrate nella scheda stessa o di comunicare con i moduli di estensione. La libreria fornisce non solo i driver Totem da installare nell'IDE, ma anche numerosi esempi funzionali, che possono essere utilizzati come trampolino di lancio. La "Libreria Totem" può essere installata come qualsiasi altra libreria dal gestore delle librerie di Arduino (Strumenti -> Gestore della libreria -> cerca "Libreria Totem"). RoboBoard utilizza il processore ESP32 utilizzato anche in numerose altre schede basate su Arduino, semplificando l'inclusione e il codice da altri progetti su RoboBoard. Un dettaglio impor-

Fig. 4
Principio
di funzionamento
del robot
line-follower.



tante è che le periferiche time-critical (ossia quelle di cui occorre la certezza temporale del controllo) come i servomotori e i motori CC, sono controllate da un processore dedicato, il che rende libero l'ESP32 di eseguire del codice utente.

PROGETTI CAMPIONE

Ora che conosciamo meglio la RoboBoard, proviamo a spiegarvi come realizzare alcuni progetti dimostrativi delle potenzialità del nostro hardware; nello specifico andremo a proporre e spiegare tre diverse tipologie di progetti.

Robot che segue la linea

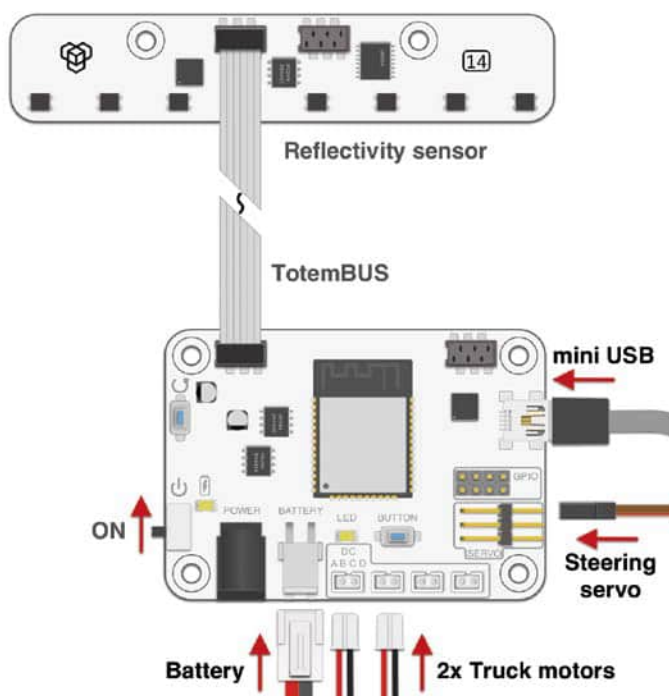
In questo progetto utilizziamo Totem TRUCK kit, una piattaforma su ruote (un telaio di camion) che lavora insieme al sensore di riflettività montato in basso sul paraurti anteriore. Quando il sensore, montato diretto verso il suolo, si sposta insieme al robot, cambia la luce riflessa in base al colore, quindi è possibile rilevare una linea e capire se la si sta seguendo o meno.

Se programiamo la RoboBoard affinché faccia sterzare il mezzo in modo che il punto a maggiore contrasto rimanga sempre nel centro, l'automezzo seguirà un percorso precostituito lungo una rotaia realizzata con un nastro di colore contrastante posato a terra. Poiché il sensore rileva il contrasto, più scuro è il materiale della striscia rispetto al suolo, meglio il robot potrà funzionare.

Per impostazione predefinita, la RoboBoard governa inizialmente solo la direzione di marcia del camion e la velocità avanti/indietro è controllata dall'app; ciò consente di mettere a punto l'algoritmo di guida e scoprire la velocità massima, il raggio di curva massimo rilevato e altri parametri.

Per la modalità di guida autonoma completa, la velocità massima dell'auto è limitata dal software ed oltre all'errore vengono gestite condizioni, come i casi in cui non viene rilevata alcuna linea; in tal caso, se il camion perde la linea torna indietro fino

Fig. 5
Schema per robot
che segue la linea.



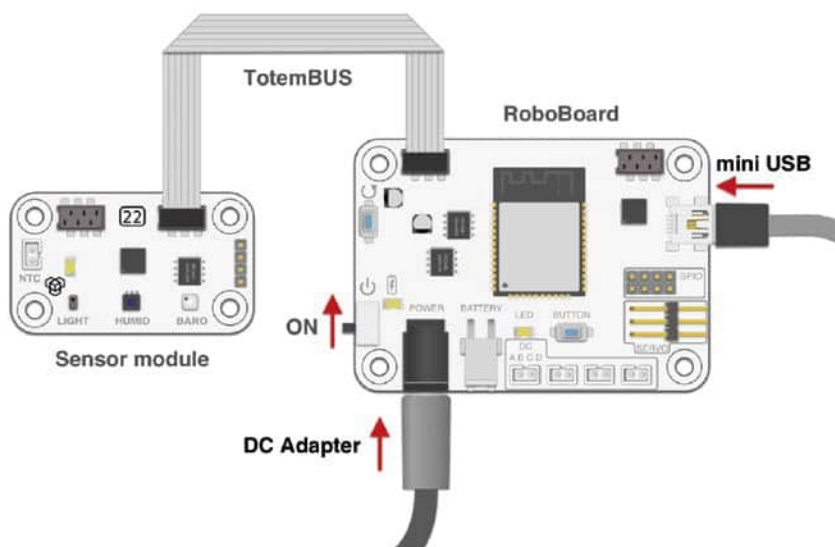


Fig. 6
Schema di collegamento per stazione meteo IoT.

a quando non viene ristabilito il rilevamento della traccia. Il codice completo per il progetto può essere trovato sul web alla pagina <https://github.com/totemmaker/RoboBoard>.

Movimento evitando le collisioni

Il progetto del robot che segue la linea può essere ampliato utilizzando un modulo di rilevamento della distanza aggiuntivo; in questo modo, il camion sarà in grado di guidare in una stanza ed evitare collisioni con pareti e altri oggetti.

Il progetto finito funziona in modo simile a un robot aspirapolvere, che rileva anche l'ambiente circostante allo scopo di muoversi nelle stanze scansando gli ostacoli.

Il modulo sensore di distanza di Totem si basa sulla misurazione del tempo di volo di un impulso laser a bassa potenza; ciò offre vantaggi rispetto ai consueti sensori a ultrasuoni, in quanto consente non solo una maggiore precisione e misurazioni di frequenza, ma anche di utilizzare diversi di questi sensori per ottenere una stima non solo della distanza dall'ostacolo, ma anche della sua forma e della direzione verso cui si trova.

In questo modo il robot sarebbe in grado di prendere una decisione migliore su come circumnavigare l'ostacolo.

In questo esempio, un singolo sensore è sufficiente per ottenere una capacità di guida autonoma. In linea di principio il metodo di funzionamento è descritto nella porzione di codice riportata qui di seguito:

```
void onDistanceMeasured(distance){
  //no close obstacles ahead
```

```
if(distance > threshold){
  driveForward();
  //we see an obstacle ahead, turn
}else if(distance > 0){
  driveLeft();
  //if we drove into an obstacle,
  run backwards
}else{
  driveBackwards();
}
}
```

Questo esempio di pseudo-codice illustra tre differenti operazioni che il robot farà in funzione delle informazioni ricevute dal sistema di misura della distanza: se la manovra non riesce a superare l'ostacolo (ad esempio un muro), e si avvicina troppo ad esso, il mezzo tornerà indietro e continuerà a girare fino a quando non avrà successo, imitando le svolte a "n" punti della vita reale.

Il codice completo può essere trovato alla pagina Github di Totem su <https://github.com/totemmaker/RoboBoard>.

Stazione meteo IoT

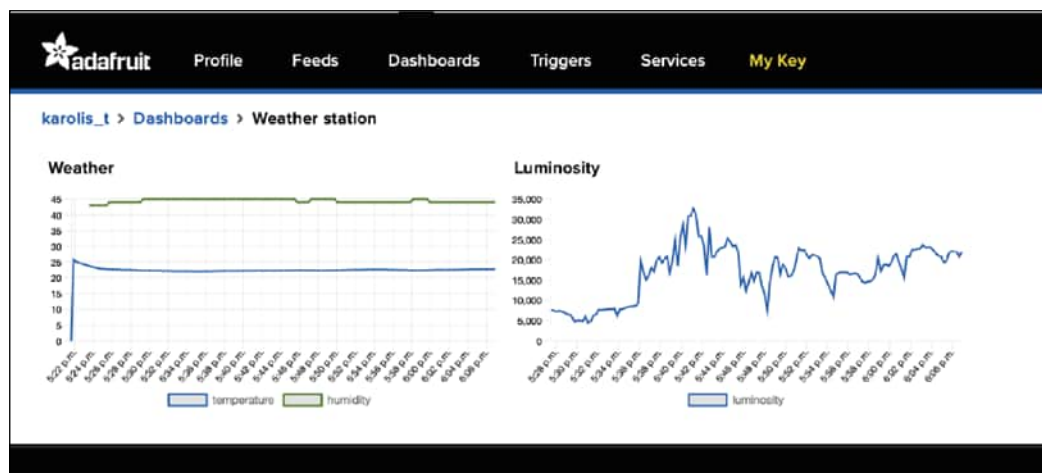
Nell'ultimo progetto di esempio, proveremo a comunicare con RoboBoard attraverso Internet; allo scopo useremo la piattaforma IO di Adafruit, anche se qualsiasi altra piattaforma del genere può andare bene. Lo schema di cablaggio corrispondente è proposto nella **Fig. 6**.

Una volta assemblato il tutto, ci collegheremo e leggeremo i dati dai sensori.

Registrando il nostro account sulla piattaforma Adafruit IO, otterremo una chiave API, che sarà



Fig. 7
Dati raccolti dal
modulo sensore sul
dashboard IO.



utilizzata da RoboBoard per caricarvi nuovi dati; dovremo inserirla insieme alle credenziali WiFi nel file **config.h**.

Il codice completo del progetto è disponibile sulla pagina del repository GitHub Totem su <https://github.com/totemaker/RoboBoard>, ma l'importante sono le parti mostrate qui di seguito, la prima delle quali serve per leggere i dati dal modulo sensore:

```
TotemModule sensor(22);
//Setup module object
//callback on module data receive
void onModuleData(ModuleData data){
    TODO
}
void setup(){
    sensor.attachOnData(onModuleData);
...

```

Quella che segue, invece, è il loop dello sketch:

```
void loop(){
    io.run() //call Adafruit IO runner
    if(millis() > (lastUpdate + IO_FREQUENCY_MS)){
        float temp = read("temperature");
        temp /= 100;
        temperature->save(temp);
        humidity->save(read("humidity"));
        luminosity->save(read("light/lumen"));
    }
}

```

La definizione **IO_FREQUENCY_MS** stabilisce quanto frequentemente inviamo nuovi valori alla piattaforma Internet.

Dopo aver caricato lo schetch sulla scheda, siamo in grado di ispezionare i nuovi valori caricati sull'in-

terfaccia web su io.adafruit.com (**Fig. 7**).

Come possiamo vedere, il lavoro richiesto per implementare nella RoboBoard la connettività IoT di base è davvero poco.

Anche se questo progetto mostra solo come caricare dati dei sensori locali su Internet, può essere modificato per utilizzarlo in una direzione diversa, permettendo di gestire la RoboBoard tramite comandi inviati da remoto attraverso il web. Questo apre a ulteriori applicazioni.

CONCLUSIONI

Come abbiamo visto dagli esempi qui proposti, RoboBoard può essere utilizzata in numerose applicazioni che vanno dall'utilizzo in semplici modelli RC fino all'implementazione di dispositivi avanzati connessi all'IoT.

Inoltre, le funzionalità del sistema possono essere sempre ampliate collegando nuovi moduli via TotemBUS, coprendo una gran quantità di applicazioni nell'ambito della robotica, dell'automazione e dell'Internet of Things. ■

Cosa occorre?

La scheda RoboBoard con ESP32 (cod. RBX4) è in vendita a Euro 49,00, il sensore di distanza (cod. RBX4DISTSENS) è disponibile a Euro 42,00, il modulo con sensori ambientali (cod. RBX4SENSMOD) costa Euro 36,00, il sensore per inseguire di linea (cod. RBX4LINFOL) è disponibile a Euro 36,00. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

OSCILLOSCOPIO 2 CANALI 70 MHZ

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

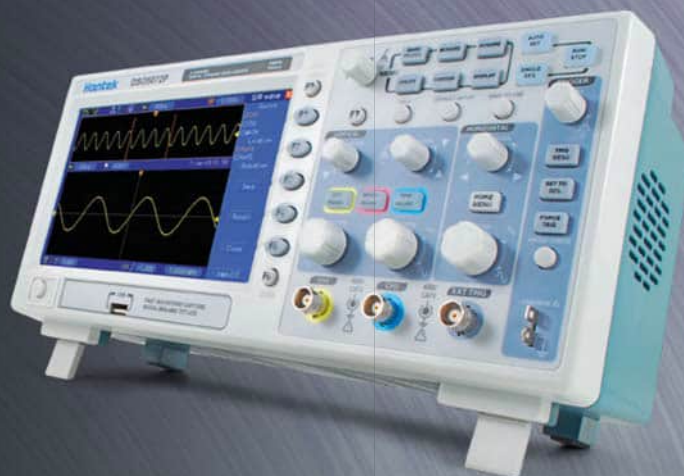
cod. DS072

€ 319,00

Disponibile anche
nella versione a 200MHz

cod. DS0202

€ 459,00



OSCILLOSCOPIO 2 CANALI 100 MHZ CON GENERATORE DI FORME D'ONDA



cod. DS0FG102

€ 429,00

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

Disponibile anche
nella versione a 200MHz

cod. DS0FG202

€ 549,00

GENERATORE DI FUNZIONI E FORME D'ONDA ARBITRARIE 2 CANALI 30 MHZ

- ✓ Frequenza onda DC 30 MHz
- ✓ 2 canali
- ✓ Frequenza di campionamento 250 MSa/s
- ✓ Risoluzione verticale 16 Bit

cod. HDG32B

€ 439,00



Disponibile anche
nella versione a 60MHz

cod. HDG62B

€ 510,00

OSCILLOSCOPIO PORTATILE 2 CANALI - 40 MHZ CON MULTIMETRO

Dispone di ampio display LCD TFT a colori da 2,8. La larghezza di banda è di 40 MHz, 250 MSa/s di velocità di campionamento in tempo reale a canale singolo. Il multimetro fornisce misurazioni di tensione continua e alternata, corrente continua e alternata, resistenze, capacità, test diodi e continuità. È interfacciabile a PC tramite software scaricabile gratuitamente. Supporta Windows 7 / 8 / 10 a 32 e 64 bit.

Prezzi IVA inclusa.

cod. MH02C42

€ 178,00



Disponibile anche nella versione
a 2 canali 70 MHz con generatore
di forme d'onda e multimetro

cod. MH02D72GEN

€ 268,00

LETTURA **GREEN PASS** CON **OPENCV** E **PYTHON**

U

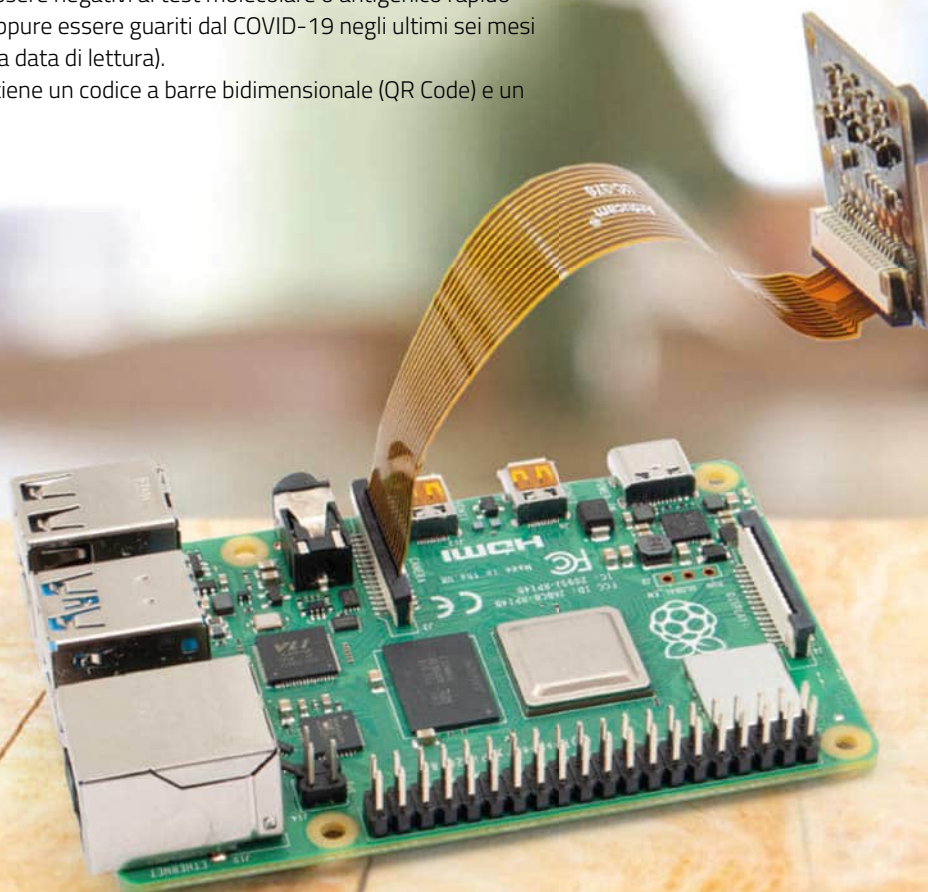
na delle misure sicuramente più dibattute negli ultimi mesi in tema di salute pubblica è il certificato verde, altrimenti noto come Green Pass. A noi che siamo elettronici non attiene la valutazione politica dello strumento e dei suoi fini, espliciti e impliciti, ma interessa l'aspetto tecnico della cosa e dietro al Green Pass c'è una tecnologia che riteniamo interessante

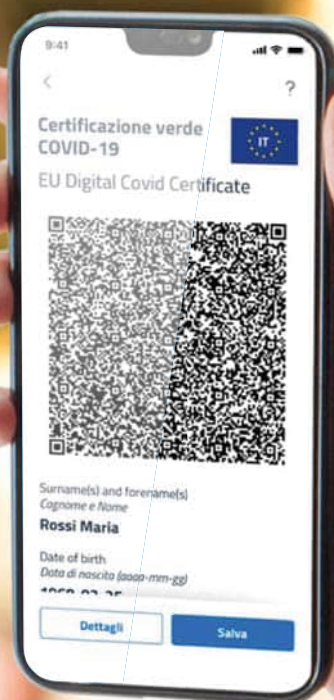
farvi conoscere e spiegare.

Il Green Pass è una certificazione digitale e stampabile (cartacea), che secondo la normativa vigente al momento della stesura di questo articolo attesta una delle seguenti condizioni: aver fatto la vaccinazione anti COVID-19 (in Italia viene emessa sia alla prima dose sia al completamento del ciclo vaccinale), essere negativi al test molecolare o antigenico rapido nelle ultime 48 ore oppure essere guariti dal COVID-19 negli ultimi sei mesi (6 mesi antecedenti la data di lettura).

La certificazione contiene un codice a barre bidimensionale (QR Code) e un

di **DANIEL SCAINI**





Leggiamo con una telecamera e Raspberry Pi il QR GreenPass mediante OpenCV e Python.

sigillo elettronico qualificato; a garanzia dell'autenticità e univocità, in Italia tale certificazione, viene emessa soltanto attraverso la piattaforma nazionale del Ministero della Salute. Lo stato italiano ha già rilasciato un'applicazione per smartphone che permette di compiere la verifica del Green Pass ma solo se coadiuvata da un operatore, il quale deve installare l'app, attivarla e fare scansioni ogni qualvolta si presenti un avventore.

A questa soluzione "manuale" potrebbe essere preferibile una struttura fissa che al passaggio della persona richieda la certificazione e che solo una volta esibita e verificata ne la validità, azioni il tornello o mostri una luce verde in modo da permettere l'entrata, senza l'intervento di alcuno. Ebbene, alla base di tale automazione potrebbe esserci qualcosa come il progetto che vi descriviamo in queste pagine, che è in grado di leggere il QR-code del Green Pass attraverso l'algoritmo specifico con il quale viene codificato; resta inteso che la lettura deve risultare trasparente a qualsiasi operatore umano, in quanto i lettori di Green Pass devono limitarsi a dare o negare il consenso all'accesso della persona che lo espone, ovvero a confermarne o meno la validità, senza entrare nel merito né rendere noti i dati personali che contiene, in ottemperanza alla c.d. Legge sulla Privacy. Prima di spiegarvi come funziona il nostro sistema, è opportuno introdurre in cosa consiste la codifica implementata nel QR-code, allo scopo di comprendere come decifrare tale codice.

LA CODIFICA

Vediamo dunque cosa contiene davvero il Green Pass: dalle specifiche rilasciate dall'Unione Europea, sappiamo che i relativi dati corrispondono a una stringa codificata **Base45** e compressa con **zlib**. Tolti i byte iniziali, possiamo risalire alla forma decodificata e decompressa.

Siamo ora davanti ad un pacchetto di dati in formato COSE, CBOR Object. Da questo pacchetto va estratto il payload che si presenterà come un classico json, che applicato allo schema fornito dall'Unione Europea (<https://raw.githubusercontent.com/ehn-dcc-development/ehn-dcc-schema/release/1.3.0/DCC.combined-schema.json>) permette di ricostruire i dati, ottenendo:

- *QR Code Issuer* : **STATO RILASCIANTE**
- *QR Code Expiry* : **DATA DI SCADENZA QR CODE**
- *QR Code Generated* : **DATA DI GENERAZIONE QR CODE**
- *Vaccination Group*

- *Dose Number* : 1
- *Marketing Authorization Holder* : **CHI HA PRODOTTO IL VACCINO (Astrazeneca, Moderna, ecc.)**
- *vaccine or prophylaxis* : **TIPO DEL VACCINO (Rna, ecc.)**
- *ISO8601 complete date: Date of Vaccination* : **GIORNO DELLA VACCINAZIONE**
- *Country of Vaccination* : **STATO IN CUI E' STATA ESEGUITA LA VACCINAZIONE**
- *Unique Certificate Identifier* : **IDENTIFICATIVO**
- *vaccine medicinal product* : EU/1/20/1528
- *Certificate Issuer* : **ENTE RILASCIANTE**
- *Total Series of Doses* : 2
- *disease or agent targeted* : **NUMERO INDICANTE LA VACCINAZIONE COVID**
- *Surname(s), forename(s)*
- *Standardised surname* : **COGNOME**
- *Surname* : **COGNOME**
- *Standardised forename* : **NOME**
- *Forename* : **NOME**
- *Schema version* : 1.0.0
- *Date of birth* : **DATA DI NASCITA**

Come si apprende ci sono dati personali limitatamente al nome del portatore e informazioni sul suo percorso diagnostico inerente al COVID-19. Con questi dati quindi è possibile sapere se il certificato, o meglio la data del certificato, è valida o meno.

Un'ulteriore prova della correttezza del certificato sarebbe validare la firma; questo pass è infatti firmato con una chiave privata dall'ente che lo rilascia. Tale firma andrebbe validata attraverso una chiave pubblica che cambia continuamente. In quest'ultimo caso sarebbe possibile identificare un Green Pass vero da uno contraffatto e dare un riscontro rispettivamente positivo o negativo durante la lettura.

IL LINGUAGGIO

Bene, stabilito ciò possiamo alla programmazione del nostro sistema. Il linguaggio di programmazione utilizzato è il Python. Orientato agli oggetti è utilizzabile per molti tipi di sviluppo software proprio per la sua facilità ed integrazione con altri tipi di linguaggio.

Chiaro e potente è paragonabile a Java o Perl, che come loro, infatti, utilizza un'elegante sintassi permettendo ai programmi di essere letti facilmente e quindi di creare progetti complessi senza rinunciare alla manutenibilità.

Oggi, secondo diversi sondaggi, è utilizzato addirittura dall'84% degli sviluppatori come linguaggio di





Fig. 1
Raspberry
Pi Imager,
il tool per
creare un SD
avviabile per
Raspberry Pi.

programmazione primario.

Per quanto riguarda gli ambiti applicativi, tra i più "gettonati" troviamo in testa la Data Analysis e Web development, ma non solo: stanno infatti cominciando ad emergere anche altri ambiti come il Machine learning, la Computer graphics ed il Game development. La versione più utilizzata è la 3.6 e spesso viene utilizzata con IDE/Editor come PyCharm Professional/Community Edition o VS Code. Dato lo scarso numero di righe che andremo a scrivere è consigliato installare un editor solo se si è alle prime armi, altrimenti i pre installati Nano o Text Editor sono più che sufficienti. Sebbene non sia ottimizzato come il C++, soprattutto per quanto riguarda l'uso di RAM e processore, è stato scelto poiché open source, con un numero veramente alto di librerie anch'esse disponibili senza costi aggiuntivi e portabile su ogni tipo di piattaforma OS (Linux, Mac, Windows). Nel nostro caso, infatti, girerà su Raspbian, che è un sistema operativo basato sui rilasci ufficiali di Debian adattati all'esecuzione su una architettura ARM, ma potrebbe tranquillamente essere eseguito su Windows senza dover apporre troppe modifiche.

RASPBERRY PI

La "macchina scelta" per l'elaborazione richiesta dal nostro progetto è senza dubbio di primaria importanza. Noi consigliamo l'uso di una Raspberry Pi 4 in modo da avere un minimo di 4GB di RAM disponibili ed anche un sistema operativo aggiornato in grado di poter supportare anche una maggiore elaborazione audio e video. La nostra board rappresenta la quarta generazione prodotta dalla Raspberry Pi Foundation e non è solo adatta a chi vuole imparare a programmare, sviluppare e giocare, ma grazie a numerosi pin di

I/O disponibili è particolarmente adatta anche per progetti complessi, ad esempio in ambito Internet of Things.

Governata da un microprocessore ARM, offre prestazioni paragonabili a quelle di un PC entry-level ma può gestire anche flussi video 4k.

Considerata ormai un caposaldo per chi appunto si cimenta in questo tipo di progetti offre numerosi dispositivi da poter abbinare come: moduli telecamera, sensori, GPS, display LCD, ecc..

Inoltre offre numerosi pin, sia di alimentazione a 3,3V, sia per poter interfacciare periferiche, relay o sensori di diversa natura e gestibili tranquillamente a livello di codice Python o C++ attraverso un insieme di librerie messe a disposizione dal produttore. Siccome per poter identificare un QR-code dobbiamo innanzitutto inquadrarlo e acquisirne l'immagine; allo scopo dobbiamo corredare la Raspberry con una PI camera rev 1.3, caratterizzata da una risoluzione di 5MP.

Questa verrà collegata direttamente sulla RPI attraverso il connettore CSI.

Questo modulo può offrire un'immagine chiara e nitida con una risoluzione di 5MP o registrare video a 1.080p con 30 fps.

Il sensore installato sul modulo ha un connettore Ribbon a 15 pin e attraverso la connessione in seriale con il connettore CSI offre un'ottima velocità di trasferimento dati, rendendo questo strumento particolarmente adatto a progetti nell'ambito embedded/mobile.

CONFIGURAZIONE

Una volta collegata la PI camera andiamo ad installare il sistema operativo all'interno della nostra Raspberry Pi. Sono necessarie tre cose per compiere questo step:

- scheda microSD-card con almeno 16/32GB;



Fig. 2
Menu del sistema operativo Raspbian.



- Raspberry Pi Imager;
- il file zip con l'immagine vera e propria del Sistema Operativo.

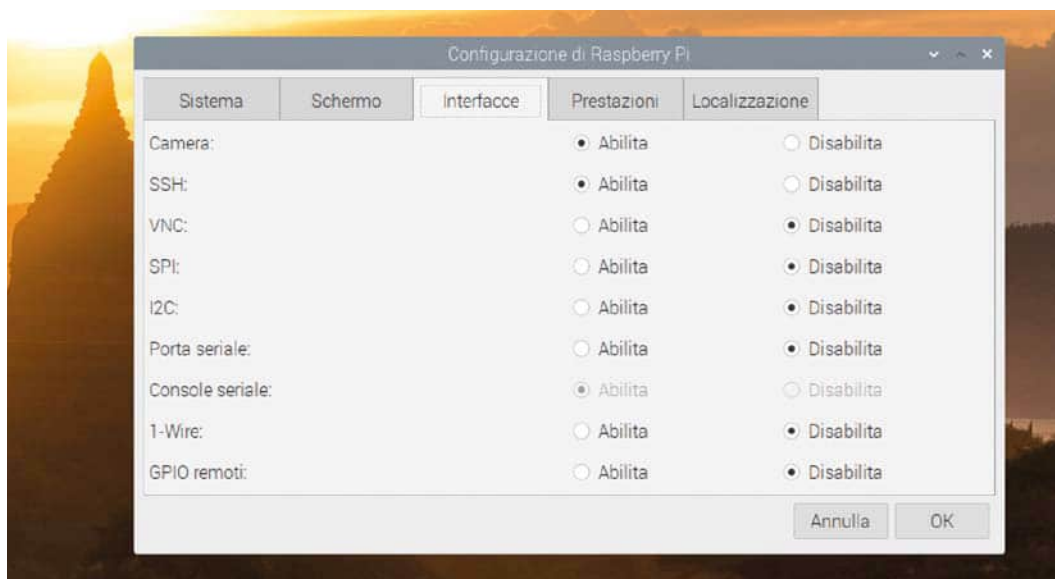
Per gli ultimi due il sito di riferimento è quello ufficiale: <https://www.raspberrypi.org/downloads/>; nella prima sezione vedremo Imager, ovvero il tool responsabile di creare un SD-card bootabile in grado di far partire il dispositivo. Più avanti, sempre all'interno del sito, troviamo invece il file .zip che andremo a caricare; per vostra comodità riportiamo il link diretto https://downloads.raspberrypi.org/raspbian_armhf_latest.

Una volta installato imager, apriamolo, inseriamo la nostra SD card, e clicchiamo CHOOSE OS (**Fig. 1**). Qui selezioniamo l'ultima voce, ovvero Custom OS e scegliamo quindi lo zip scaricato in precedenza. Successivamente clicchiamo su CHOOSE SD CARD e selezioniamo l'unità sulla quale desideriamo riportare il sistema operativo e infine WRITE. Dopo alcuni istanti avremo la nostra SD card e saremo pronti per partire con il nostro dispositivo. Notate che è possibile utilizzare l'immagine pre-configurata *rpi_32bit_greenpass.zip* all'interno della quale sono già presenti tutte le librerie e i moduli per poter utilizzare correttamente i programmi. In tal caso le credenziali per accedervi sono:

user = *pi*
password = *12345678*.

Inseriamo dunque la card all'interno del socket della nostra Raspberry Pi, colleghiamo quest'ultima ad una rete che disponga dell'accesso ad Internet ed alimentiamola. Una volta seguite le prime indicazioni sulle impostazioni generali come password, rete ecc, andiamo dentro alle impostazioni, cliccando sull'icona con il lampone in alto a sinistra e, dal menu corrispondente (mostrato nella **Fig. 2**) scegliamo Preferenze > Configurazione Raspberry Pi. Apparirà una schermata come quella mostrata in figura e quindi clicchiamo su Interfacce e abilitiamo le prime due voci, ovvero Camera e SSH (**Fig. 3**). La prima per ovvi motivi, nel caso in cui volessimo appunto usare un modulo camera Pi esterno, mentre la seconda per poterci connettere

Fig. 3
Schermata di configurazione delle periferiche esterne come SSH o Fotocamera.



in remoto alla macchina attraverso tool terminale come PuTTY o WinSCP.

Chiudiamo il menu cliccando su OK e ci chiederà di riavviare la macchina: procediamo.

Al riavvio apriamo un terminale e sempre per poter raggiungere la macchina da remoto installiamo il desktop remoto impartendo il comando:

```
sudo apt-get install xrdp -y
```

Completata l'installazione, provvediamo all'aggiornamento dei pacchetti del sistema operativo:

```
sudo apt-get update
```

Fatto ciò andiamo a impartire il comando:

```
sudo apt-get upgrade
```

per completare l'aggiornamento del sistema. Ora abbiamo tutti gli strumenti per raggiungere la macchina da remoto e quindi possiamo dimenticarci del monitor e agire direttamente dal nostro PC: a voi la scelta. Installiamo ora il framework indispensabile per ogni progetto di visione artificiale: OpenCV. OpenCV, come già descritto in articoli precedenti, è una libreria ed un insieme di algoritmi scritti in C per lo streaming e la manipolazione image/video real-time per la visione artificiale disponibile ormai per tutte le piattaforme OS. Questa permette la trattazione di immagini in maniera semplice, gestendole come matrici di pixel; in breve la libreria ci consente di effettuare quel processo che sta alla base della computer vision, ovvero l'Image processing, andando a modificare/migliorare le immagini o le sequenze di immagini (video). È molto utilizzata in ambito scientifico e di ricerca e può essere adoperata in diversi settori: controllo della qualità, ricerca, difesa e molti altri. Con poche righe di istruzioni, attraverso questa libreria è possibile riconoscere le persone all'interno di una foto, leggere in maniera automatica delle targhe, azioni che altrimenti sarebbe molto complicato implementare e integrare in progetti più ampi. Ovviamente nel caso di dispositivi embedded come questo l'installazione non può essere lanciata con un solo comando, ma per avere un controllo puntuale della versione dovremmo andare ad installarla partendo direttamente dai sorgenti. Per far ciò apriamo nuovamente il terminale e dalla shell digitiamo i comandi riepilogati qui di seguito:

```
sudo apt-get install build-essential cmake git unzip
```

```
pkg-config -y
sudo apt-get install libjpeg-dev libpng-dev libtiff-dev -y
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev -y
sudo apt-get install libgtk2.0-dev libcanberra-gtk*-y
sudo apt-get install libxvidcore-dev libx264-dev libgtk-3-dev -y
sudo apt-get install python3-dev python3-numpy python3-pip -y
sudo apt-get install python-dev python-numpy -y
sudo apt-get install libtbb2 libtbb-dev libdc1394-22-dev -y
sudo apt-get install libv4l-dev v4l-utils -y
sudo apt-get install libjasper-dev libopenblas-dev libatlas-base-dev libblas-dev -y
sudo apt-get install liblapack-dev gfortran -y
sudo apt-get install gcc-arm* -y
sudo apt-get install protobuf-compiler -y
```

Avete così provveduto all'installazione delle varie librerie che occorrono per la compilazione di OpenCV.

Queste ci permetteranno di installare tutti quei componenti di sistema che ci renderanno autonomi nel compilare i sorgenti e di compiere l'installazione del caso. Scarichiamo dunque i sorgenti e decomprimiamoli innanzitutto cambiando directory con il comando:

```
cd ~
```

per raggiungere la HOME. Poi i due comandi:

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.2.0.zip
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.2.0.zip
```

per effettuare il download dei pacchetti OpenCV. Fatto ciò, con i comandi seguenti si aprono i file zip, li si scompattano e si rinominano le cartelle:

```
unzip opencv.zip
unzip opencv_contrib.zip
mv opencv-4.2.0 opencv
mv opencv_contrib-4.2.0 opencv_contrib
```

Installiamo l'ambiente virtuale dove poter compilare senza fare danni o alterare le dipendenze dell'intero sistema operativo; per prima cosa impartiamo il comando:



```
sudo pip3 install virtualenv
```

per effettuare l'installazione dell'ambiente virtuale (Fig. 4). Impartiamo quindi i comandi:

```
sudo pip3 install virtualenvwrapper
echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3.7" >> ~/.bashrc
```

per l'export di alcuni path fondamentali. Impartiamo quindi i comandi:

```
echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.bashrc
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc
source ~/.bashrc
```

Poi con il comando:

```
mkvirtualenv cv420
```

creiamo l'ambiente e vi entriamo (questo passaggio è opzionale).

Passiamo poi ad impartire i comandi seguenti:

```
pip3 install numpy
```

per installare la libreria python numpy.

Poi, con il comando:

```
cd opencv
```

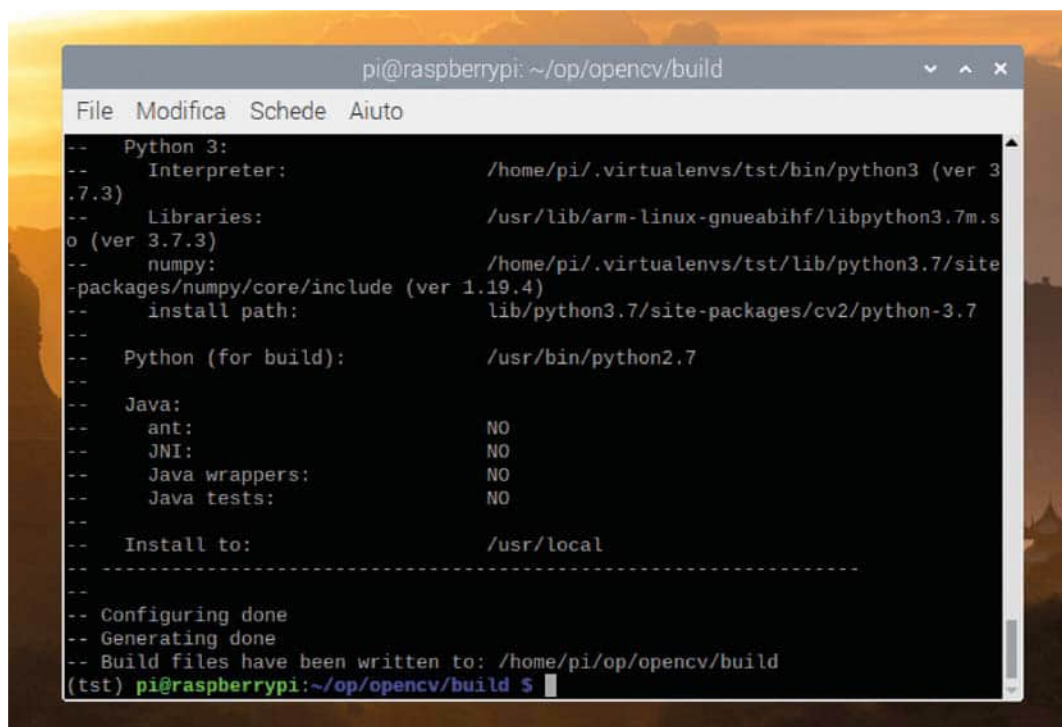
entriamo nella cartella di OpenCV scaricata e rinominata nei passi precedenti:

```
mkdir build
cd build
```

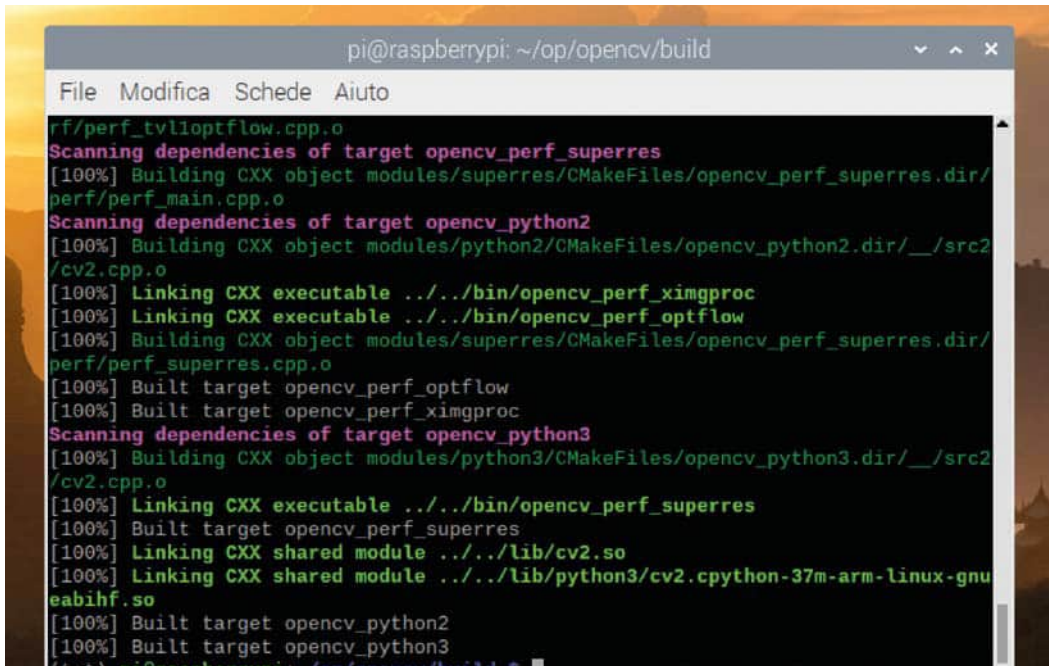
A questo punto procediamo con questi comandi:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE
-D CMAKE_INSTALL_PREFIX=/usr/local
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules
-D ENABLE_NEON=ON -D ENABLE_VFPV3=ON
-D WITH_OPENMP=ON -D BUILD_TIFF=ON
-D WITH_FFMPEG=ON -D WITH_GSTREAMER=ON
-D WITH_TBB=ON -D BUILD_TBB=ON
-D BUILD_TESTS=OFF -D WITH_EIGEN=OFF
-D WITH_V4L=ON -D WITH_LIBV4L=ON
-D WITH_QT=OFF -D WITH_VTK=OFF
-D OPENCV_EXTRA_EXE_LINKER_FLAGS=-latomic
-D OPENCV_ENABLE_NONFREE=ON
-D INSTALL_C_EXAMPLES=OFF
-D INSTALL_PYTHON_EXAMPLES=OFF
-D BUILD_NEW_PYTHON_SUPPORT=ON
-D BUILD_opencv_python3=TRUE
-D OPENCV_GENERATE_PKGCONFIG=ON
```

 **Fig. 4**
Generazione con successo della configurazione per la compilazione di OpenCV.



```
pi@raspberrypi: ~/op/opencv/build
File Modifica Schede Aiuto
-- Python 3:
-- Interpreter: /home/pi/.virtualenvs/tst/bin/python3 (ver 3.7.3)
-- Libraries: /usr/lib/arm-linux-gnueabi/libpython3.7m.so (ver 3.7.3)
-- numpy: /home/pi/.virtualenvs/tst/lib/python3.7/site-packages/numpy/core/include (ver 1.19.4)
-- install path: lib/python3.7/site-packages/cv2/python-3.7
--
-- Python (for build): /usr/bin/python2.7
--
-- Java:
-- ant: NO
-- JNI: NO
-- Java wrappers: NO
-- Java tests: NO
--
-- Install to: /usr/local
-----
-- Configuring done
-- Generating done
-- Build files have been written to: /home/pi/op/opencv/build
(tst) pi@raspberrypi:~/op/opencv/build $
```



```
pi@raspberrypi: ~/op/opencv/build
File Modifica Schede Aiuto
rf/perf_tv11optflow.cpp.o
Scanning dependencies of target opencv_perf_superres
[100%] Building CXX object modules/superres/CMakeFiles/opencv_perf_superres.dir/
perf/perf_main.cpp.o
Scanning dependencies of target opencv_python2
[100%] Building CXX object modules/python2/CMakeFiles/opencv_python2.dir/__/src2
/cv2.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_ximgproc
[100%] Linking CXX executable ../../bin/opencv_perf_optflow
[100%] Building CXX object modules/superres/CMakeFiles/opencv_perf_superres.dir/
perf/perf_superres.cpp.o
[100%] Built target opencv_perf_optflow
[100%] Built target opencv_perf_ximgproc
Scanning dependencies of target opencv_python3
[100%] Building CXX object modules/python3/CMakeFiles/opencv_python3.dir/__/src2
/cv2.cpp.o
[100%] Linking CXX executable ../../bin/opencv_perf_superres
[100%] Built target opencv_perf_superres
[100%] Linking CXX shared module ../../lib/cv2.so
[100%] Linking CXX shared module ../../lib/python3/cv2.cpython-37m-arm-linux-gnu
eabi9hf.so
[100%] Built target opencv_python2
[100%] Built target opencv_python3
```

 **Fig. 5**
Compilazione
di OpenCV.

```
-D BUILD_EXAMPLES=OFF ..
```

e compiliamo la libreria con il comando (Fig. 5):

```
make -j4
```

Vi facciamo notare che i relativi tempi di compilazione possono variare in base alla potenza di calcolo del computer su cui stiamo lavorando; nel nostro caso abbiamo utilizzato un Raspberry Pi 4 equipaggiata con 8GB di RAM e ci abbiamo messo circa una mezz'oretta.

Ora installiamo il compilato con i seguenti comandi:

```
sudo make install
```

con il quale provvediamo all'installazione del compilato del passaggio precedente e:

```
sudo ldconfig
```

Poi passiamo all'aggiornamento dei pacchetti di sistema operativo con:

```
sudo apt-get update
```

Poi impartiamo i comandi:

```
cd ~
rm opencv.zip
rm opencv_contrib.zip
```

per effettuare la rimozione degli zip scaricati.

Fatto ciò impartiamo i seguenti comandi:

```
cd ~/.virtualenvs/cv420/lib/python3.7/site-packages
ln -s /usr/local/lib/python3.7/site-packages/cv2/py-
thon3.7/cv2.cpython-37m-arm-linux-gnueabi9hf.so
cd ~
cd ~/opencv/build/lib/python3
sudo cp cv2.cpython-37m-arm-linux-gnueabi9hf.so /usr/
local/lib/python3.7/dist-packages/cv2.so
cd ~
```

Con cui creiamo link simbolici per l'ambiente virtuale (se creato).

Ora installiamo le librerie responsabili della decompressione, della decodifica in Base45 e infine per la lettura dei barcode e qrcode.

A seconda che si voglia o meno modificare il nostro codice di partenza, un passaggio opzionale potrebbe essere quello di non usare OpenCV per quest'ultima feature. Infatti, nonostante questo offra già al suo interno diversi metodi per la lettura dei barcode bidimensionali e monodimensionali, una delle librerie più utilizzate è sicuramente Pyzbar, un porting ad-hoc della famosissima zbar.

Andiamo ad installarla ugualmente per poter eventualmente estendere i nostri sviluppi futuri.

Ultimiamo questa fase con i seguenti comandi:

```
pip3 install pyzbar
```

necessario all'installazione della libreria per bar-

code e:

```
pip3 install cbor2
pip3 install base45
pip3 install cose
```

per l'installazione librerie per la decompressione e la decodifica in base45.

A questo punto scarichiamo i sorgenti Python:

```
git clone https://github.com/danisk89/greenPassLitePython.git
cd greenPassLitePython
```

Per poter lanciare uno dei due applicativi al suo interno digitiamo:

```
python3 consoleqr.py
```

Per la versione senza UI, oppure:

```
python3 qrcode.py
```

Per la versione con interfaccia nel quale si vede ciò che viene inquadrato dalla camera.

IL CODICE

Entrando nella cartella sopracitata vediamo tre file:

dec_green.py, **consoleqr.py** ed infine **qrcode.py**.

Esaminiamoli in quest'ordine cercando di capire i punti salienti di ogni file.

Il primo, **dec_green**, si può notare che riporta la classe di decodifica vera e propria del greenpass. La funzione cardine, che verrà poi chiamata dall'esterno è la **main**, definita in fondo nel file:

```
def main(self, qrString):
    #print(qrString)
    result = Result()
    certificate = Certificate();
```

Questa come si può osservare prende in ingresso una stringa, la quale può essere passata da una camera, come in questo caso, oppure da un lettore di codici a barre sulla seriale.

Quest'ultimo, così come tutti i prodotti citati, è disponibile nello store on-line di Futura Elettronica, www.futurashop.it.

Il primo parametro invece è sempre presente nelle funzioni di una classe python per poter usare metodi o parametri interni alla classe stessa.

Come primo passo viene valutato l'header perché sia conforme con la codifica del certificato verde digitale, ovvero HCI.

```
qrString = self.checkHeader(qrString)
```

Poi viene eseguita la decodifica e la decompressione del pacchetto:

```
try:
    zlibString = self.base45Decode(qrString)
    #2.
    try:
        coseString = self.zlibUnpack(zlibString)
        #3.
        try:
            decoded = self.
COSEdecompression(coseString)
```

Ed infine si arriva alla trasformazione della stringa json ottenuta in una classe vera e propria la quale sarà poi ritornata al chiamante.

```
payload = decoded.payload
try:
    decodedCbor = self.cborToJson(payload)
    #5.
    try:
        certificate = self.
jsonToClass(decodedCbor)
```



La scheda Raspberry Pi equipaggiata con la PI camera.



Listato 1

```
while True:
    frame = vs.read()
    if frame is not None:
        bs = decode(frame)
        for bar in bs:
            if modeVideoDate is None:
                bdata = bar.data.decode("utf-8")
                btype = bar.type
                modeVideoDate = datetime.datetime.now()
                res = getGreen.main(bdata)
                print(res)
            if modeVideoDate is not None:
                now = datetime.datetime.now()
                past = modeVideoDate + datetime.timedelta(0, 5)
                if datetime.datetime.timestamp(past) <= datetime.datetime.timestamp(now) :
                    modeVideoDate = None
                    print("cancel")

fps.stop()
cv2.destroyAllWindows()
vs.stop()
```

```
except:
    result.errorCode = 1004
    result.errorDesc = "Error on JSON
```

to class conversion"

Questi sono spezzoni del codice; il listato intero, per ragioni di spazio lo rimandiamo al nostro sito, dal quale potete scaricarlo insieme agli altri file della rivista.

Passiamo ora al secondo, ovvero **consoleqr.py**, il primo dei due applicativi che sfruttano la precedente classe, ma senza interfaccia grafica. Al suo interno troviamo la classica struttura per poter far girare OpenCV ed acquisire frame dalla propria camera:

```
vs = VideoStream(src=0,usePiCamera=True,resolu-
tion=(1280,720)).start()
fps = FPS().start()
```

Si può notare una prima parte nella quale viene istanziata la camera con una risoluzione piuttosto alta.

Questo per poter sfruttare meglio il device e ricevere in ingresso alla prossima istruzione un'immagine molto grande da poter valutare sicuramente meglio il QR-code in essa contenuto.

```
detector = cv2.QRCodeDetector()
modeVideoDate = None
getGreen = DecodeGreen()
```

Istanziamo qui la classe di OpenCV per i barcode e la classe di cui abbiamo già parlato precedente-

mente responsabile della decodifica.

Creiamo ora il solito ciclo per la lettura dei frame: lo facciamo con la porzione di codice contenuta nel

Listato 1.

È stata aggiunta una data per poter evitare di leggere in continuazione il dato, ma una volta ottenuta una scansione valida, il sistema entra in pausa per 5 secondi.

Il codice completo lo trovate, come gli altri, tra i file del progetto sul nostro sito.

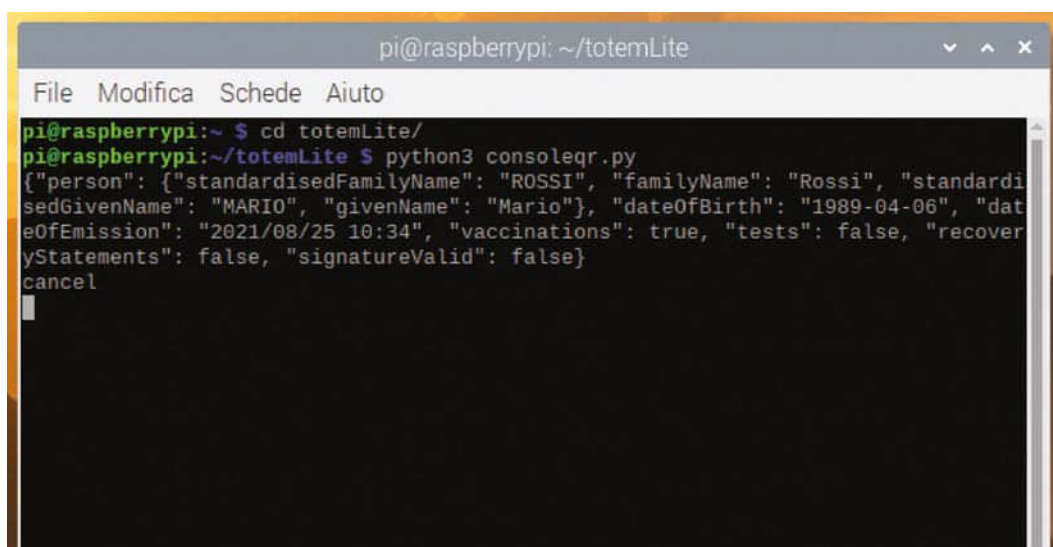
Infine nell'ultimo file **qrcode.py** vi è esattamente lo stesso programma, con l'aggiunta di un frame per poter visionare cosa si sta inquadrando. Infatti nelle righe finali viene aggiunta questa porzione di codice:

```
if modeVideoDate is not None:
    now = datetime.datetime.now()
    past = modeVideoDate + datetime.time-
    delta(0, 5)
    if datetime.datetime.timestamp(past) <=
    datetime.datetime.timestamp(now) :
        modeVideoDate = None
        cv2.imshow("Frame", frame)
        if cv2.waitKey(10) & 0xFF == ord(mySettings.
        keyToExit):
            break
        fps.update()
    fps.stop()
cv2.destroyAllWindows()
vs.stop()
```

Questa parte di codice è responsabile della cre-



➔ **Fig. 6**
Risultato della
scansione di un
Green Pass.



```
pi@raspberrypi: ~/totemLite
File Modifica Schede Aiuto
pi@raspberrypi:~ $ cd totemLite/
pi@raspberrypi:~/totemLite $ python3 consoleqr.py
{"person": {"standardisedFamilyName": "ROSSI", "familyName": "Rossi", "standardisedGivenName": "MARIO", "givenName": "Mario", "dateOfBirth": "1989-04-06", "dateOfEmission": "2021/08/25 10:34", "vaccinations": true, "tests": false, "recoveryStatements": false, "signatureValid": false}}
cancel
```

azione di un frame che permette di vedere i vari frame che vengono letti dal device. Il codice completo lo trovate sempre sul nostro sito www.elettronica.in.it. Nella **Fig. 6** è proposta la schermata che riporta i vari frame letti dal nostro dispositivo in un QR-code di Green Pass.

CONCLUSIONI

Il progetto che vi abbiamo proposto in queste pagine costituisce una buona base di partenza per poter approssicare la lettura del certificato verde. Come accennato in precedenza vi sono diverse verifiche da effettuare per validare i dati letti, non è infatti solamente sufficiente acquisire la data di vaccinazione.

Infatti in questo esempio non è stata presa in considerazione la verifica della firma digitale, importantissima per distinguere un pass vero da uno contraffatto. Un impostore infatti potrebbe tranquillamente copiare il blocco di un'altra persona, cambiare il nome, il cognome o la data di nascita ed il gioco è fatto.

La firma, la cui chiave pubblica cambia di continuo, è un aspetto fondamentale, infatti è questo uno dei motivi che non permette un offline prolungato del sistema.

Inoltre al momento non sono state attivate politiche per l'autofocus, quindi anche la cattura dell'immagine può avere una sua tempistica e quindi portare a rallentare il sistema.

Dalla schermata di configurazione di Raspbian si potrebbero abilitare i GPIO presenti sulla scheda di Raspberry e attraverso il codice Python, abilitarli o

meno a seconda del risultato della verifica. In questo modo si potrebbero andare a comandare relé collegati a sistemi di ingresso o luci a LED di colore diverso in base all'esito dell'analisi dei dati del codice e relativo confronto. Infine facciamo notare che in questo articolo è stato usato Raspbian a 32bit come sistema operativo, ma nulla ci vieta di passare alla versione a 64bit oppure di usare la versione Server; infatti, come abbiamo visto con l'esempio "consoleqr.py" non è necessaria la parte di UI (Interfaccia utente). ▣



Cosa occorre?

La scheda Raspberry Pi 4 è disponibile in diverse versioni da Futura Elettronica. La versione con 8GB di memoria (cod. RPI4-8GB) è in vendita a Euro 99,00, la versione con 4GB di memoria (cod. RPI4-4GB) è disponibile a Euro 69,00, la stessa versione con 2GB di memoria (cod. RPI4-2GB) è in vendita a Euro 59,00. La telecamera da 5Mpx utilizzata in questo progetto (cod. YG016) è in vendita a Euro 9,50.

Per le board Raspberry Pi 4 sono disponibili anche diversi starter kit già pronti all'utilizzo disponibili alla pagina bit.ly/RPIstarterKit

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

Analizza i circuiti nei minimi dettagli!

Completa il tuo laboratorio di elettronica con le lampade a LED con lente d'ingrandimento e realizza i tuoi progetti in modo più semplice e preciso.

Lampada LED con lente d'ingrandimento 3 diottrie - 1.200 lumen



Lampada costituita da 60 LED bianchi con lente d'ingrandimento Ø127 mm a 3 diottrie con flusso luminoso di 1.200 lumen. Particolarmente indicata per compiere lavori di precisione. Utile anche per uso domestico, dispone di regolazione della luminosità su quattro livelli, un doppio braccio che ne permette l'orientamento nella posizione più adatta. Fornita di morsetto metallico per il fissaggio al banco. Alimentazione 220 VAC. Colore bianco. Consumo 14 W.



€78,00
Cod. LAMPDESK3

DISPONIBILI ANCHE NELLE
VERSIONI A 5 DIOTTRIE
COD. LAMPDESK5 € 78,00
E A 8 DIOTTRIE
COD. LAMPDESK8 € 78,00

Lampada professionale a LED dimmerabile - 880 lumen



Lampada da tavolo professionale dimmerabile, dotata di 90 LED SMD. Fornita di morsetto metallico per il fissaggio al banco, dispone di doppio braccio con una lunghezza di 400 mm. Alimentazione 220 VAC - 240 VAC / 50 Hz. Colore bianco. Offre la possibilità di variare la luminosità tramite i pulsanti touch.

€82,00
Cod. TP233

Lampada a LED con lente d'ingrandimento 5 diottrie 600 lumen

Lampada costituita da 30 LED bianchi con lente d'ingrandimento Ø100 mm a 5 diottrie. Ideale per lavorazioni di precisione, può essere posizionata ovunque grazie al braccio flessibile. L'elevato peso della base garantisce una eccellente stabilità. Consumo 7,5 W. Alimentazione 12 VDC, tramite adattatore da rete 230 VAC / 50 Hz.



€39,00
Cod. LAMP9101AE

Prezzi IVA inclusa.

I tuoi PCB a partire da

1€

Con il servizio PCB di Futura Elettronica, potrai realizzare facilmente il circuito stampato del tuo progetto!

Vai su www.futurashop.it, inserisci nella ricerca il codice "PCBPRODUCTION" e carica il tuo Gerber scegliendo le dimensioni e le caratteristiche preferite.

Prezzi IVA inclusa.



cod. PCBPRODUCTION

COLORE SERIGRAFIA



BIANCA



NERA

La serigrafia nera è disponibile solamente per PCB bianchi e gialli.

COLORE SOLDER



VERDE

ROSSO

GIALLO

BLU

BIANCO

NERO

SPESSORE (mm)



0.2

0.4

0.6

0.8

1.0

1.2

1.6

2.0

2.4

2.8

3.2

NUMERO DI LAYERS

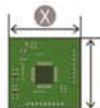
1 Layer

2 Layers

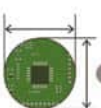
4 Layers

6 Layers

DIMENSIONI (cm)



Single Size



Pannel Size

- 5x5 - 5x10 - 5x15 - 5x20 - 5x25 - 5x30
- 10x10 - 10x15 - 10x20 - 10x25 - 10x30
- 15x15 - 15x20 - 15x25 - 15x30
- 20x20 - 20x25 - 20x30
- 25x25 - 25x30
- 30x30

ECCO QUALCHE ESEMPIO:

PCB con 1 layer
da 5x5 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 1 cad.

PCB da 2 layer
da 10x10 cm
spessore 1,6 mm
solder verde
serigrafia bianca
€ 6 cad.

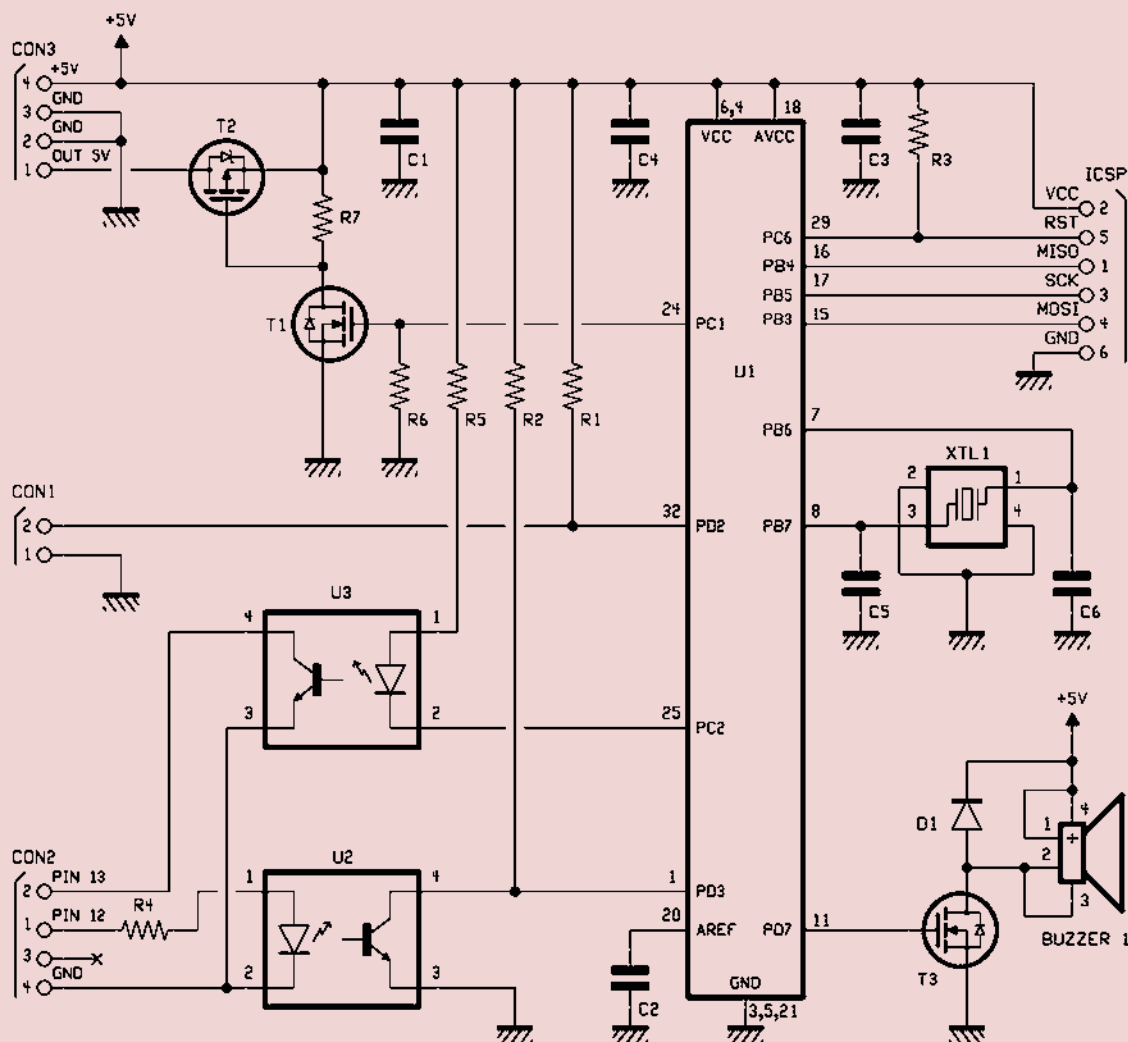
MIRROR COPPO



di MAURIZIO ŠKERLIČ

MirrorCoppo è uno specchio Smart realizzato con l'assistente vocale Amazon Alexa installato sulla Raspberry Pi, controllato da vocale o gesture. Seconda e ultima puntata.

Del nostro specchio Smart vi abbiamo spiegato sinora le funzionalità e gli aspetti della programmazione. Adesso è giunto il momento di passare alla parte pratica, analizzandone l'hardware e spiegando in che modo le varie unità (schede elettroniche e sensori) devono essere interconnessi per pervenire al risultato finale. Oltre alla Raspberry Pi, il progetto MirrorCoppo ha bisogno di diversi circuiti che sono stati raggruppati in due schede: la scheda interruttore, cioè quella per l'accensione e lo spegnimento controllato della Raspberry Pi, visibile in **Fig. 1**, e la scheda principale visibile in **Fig. 2**, composta dal microcontrollore, che gestisce l'accensione delle strisce a LED RGB e del monitor, dagli amplificatori, dall'orologio in tempo reale RTC (Real Time Clock) visibile in **Fig. 3**, dal sensore di temperatura e umidità HTU21D e dal sensore controllo gesti APDS9960



visibile in **Fig. 4**, a catalogo Futura elettronica con il codice APDS9960BOUT.

SCHEMA ELETTRICO SCHEDA INTERRUTTORE

Incominciamo con l'analizzare il circuito elettrico della scheda interruttore. Il circuito utilizza lo stes-

so microcontrollore a 8 bit ATMEGA328P montato sulla scheda Arduino Uno, il quale si interfaccia con la Raspberry Pi mediante due foto accoppiatori. Abbiamo optato per l'utilizzo dei foto accoppiatori per due motivi, il primo per convertire i livelli logici delle porte I/O e il secondo per disaccoppiare galvanicamente le porte e proteggerle da eventuali interferenze o picchi di tensione. Al piedino 32 di U1 viene collegato un pulsante esterno che fa capo al connettore CON1, la cui pressione viene segnalata da un breve segnale acustico emesso dal buzzer BZ1. Quest'ultimo è pilotato dal MOSFET a canale N siglato T3 il cui gate è collegato al piedino 11 del microcontrollore U1. Una pressione prolungata del pulsante, segnalata da un segnale acustico più lungo, mette il piedino 24 di U1 a livello alto portando in conduzione il transistor MOSFET T1 a canale N. Questo a sua volta mette in conduzione il secondo transistor MOSFET di potenza T2 a canale P che, come un interruttore, fa scorrere la corrente



Fig. 1
Scheda interruttore.

attraverso il piedino 1 del connettore CON3, accendendo la Raspberry Pi collegata. Viceversa se il pulsante viene premuto per cinque secondi, il buzzer BZ1 emette un segnale acustico più lungo e il piedino 25 di U1 va basso, facendo partire la sequenza di shutdown della Raspberry Pi. Alla fine della sequenza la RPI pone a livello basso il piedino 12 del connettore CON3, avvisando che è pronta per essere spenta in sicurezza. Questo segnale porta in conduzione il diodo led del foto accoppiato U2 portando a livello basso il piedino 1 di U1. In risposta a questo segnale il microcontrollore U1 ATMEGA328P porta basso il piedino 24, interdicendo così la corrente alla Raspberry Pi spegnendola. Il MOSFET T2 da noi scelto ha una resistenza Drain Source (abbreviato RDSon) molto bassa per garantire una bassa caduta di tensione sulla linea a 5 Volt. Non sostituitelo con un componente la cui RDSon è maggiore, altrimenti a causa della caduta di tensione la Raspberry Pi non funzionerà correttamente. Quando la tensione di alimentazione è troppo bassa la Raspberry Pi lo segnala visualizzando nell'angolo alto destro un'icona gialla a forma di saetta.

SCHEMA ELETTRICO SCHEDA PRINCIPALE

Analizziamo ora il circuito elettrico della seconda scheda, quella principale, il cui cuore è il microcontrollore Espressive ESP32 dual core a 80MHz. Il suo compito è quello di controllare l'accensione delle strisce LED RGB mediante tre transistor Darlington T1, T2 T3, controllare l'accensione del monitor mediante i transistor T4 e T5 e comunicare, mediante il protocollo I²C, con i sensori APDS-9960 e HTU21D, nonché interfacciarsi alla RPI mediante la porta USB ed inviare i dati di temperatura e umidità al modulo "MMM-ArduPort" per essere visualizzati a monitor dal MagicMirror. Sullo stampato trovano posto anche una coppia di amplificatori da 2,5 W in classe D siglati PA-M8302A, a catalogo Futura Elettronica codice PA-M8302A, alla cui uscita vanno collegati una coppia di diffusori da 3 W e 4 ohm come quelli a catalogo Futura Elettronica codice AP5243W. Questi moduli possiedono al loro interno un circuito di shutdown che fa capo al piedino 3, il quale se messo a basso livello spegne l'amplificatore. Sfruttando questa caratteristica abbiamo interfacciato questo pin a un circuito anti-bump cioè un circuito che sopprime il fastidioso botto udibile dai diffusori all'accensione di un amplificatore. Questo semplice circuito è costituito da due transistor NPN T6 e T7,

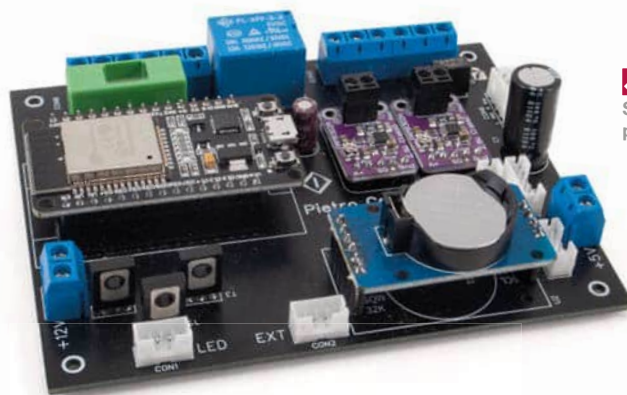


Fig. 2
Scheda
principale.

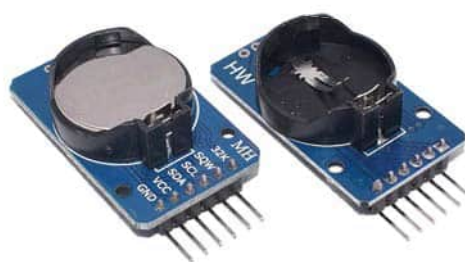
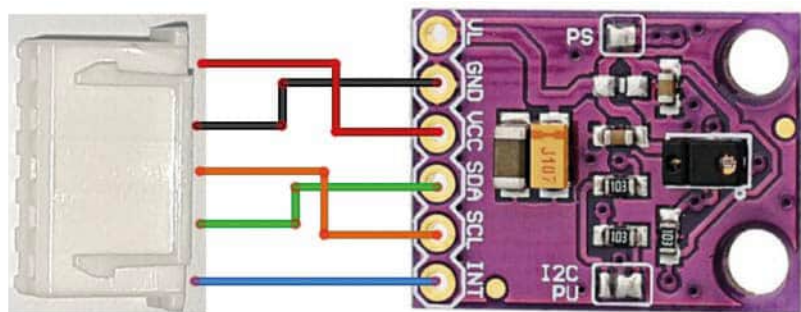


Fig. 3
Modulo
RTC codice
DS3231RTC con
e senza batteria
CR2032.

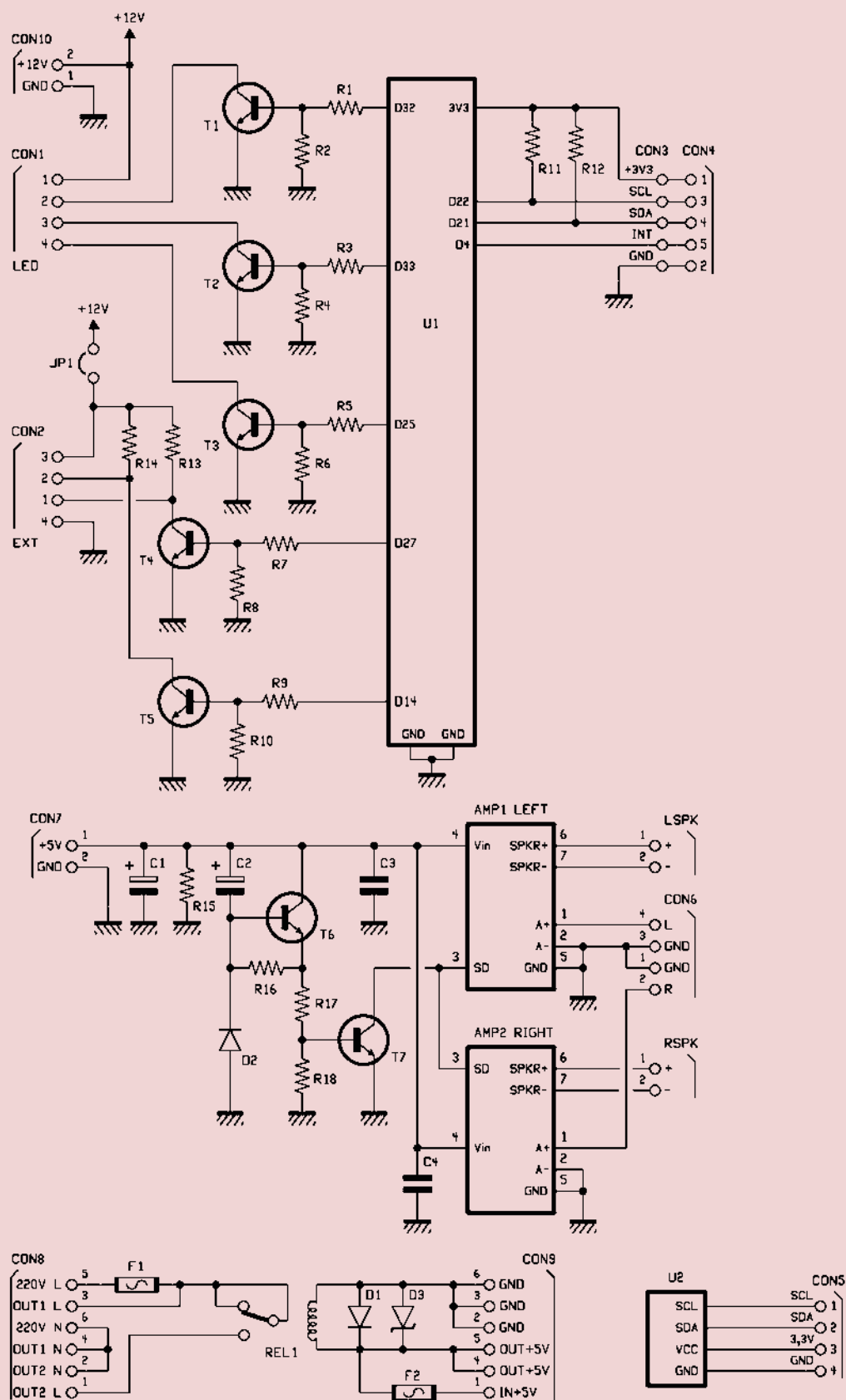
dal condensatore C2, dal diodo D2 e da quattro resistenze R15, R16, R17 e R18. Il transistor T6 è un inseguitore di emettitore che come carico sull'emettitore utilizza le resistenze R17 e R18, inoltre ha una resistenza R16 collegata tra la base e l'emettitore, questo fa sì che l'impedenza di R16, vista dalla base di T6, sia parecchie volte maggiore del valore della resistenza in corrente continua. L'impedenza in ingresso composta dal condensatore C2 e dal transistor T6, uguale all'impedenza in parallelo della resistenza R16 e della base di T6, formano un circuito temporizzatore RC. Quando viene data corrente per la prima volta al circuito, il condensatore elettrolitico C2 è completamente scarico e mette in conduzione T6 facendo passare la corrente positiva, che attraverso la resistenza R17 porta in conduzione il transistor T7, in

Fig. 4
Sensore
APDS9960.
La piedinatura del
sensore è diversa
da quella della
scheda principale:
la posizione dei
piedini GND, VCC
e SDA, SCL
è invertita.



schema elettrico

SCHEDA PRINCIPALE



configurazione di emettitore comune, portando a sua volta a livello basso i piedini 3 degli amplificatori PAM8302A spegnendoli. Il condensatore C2 inizia a caricarsi tramite l'impedenza formata dalla resistenza R16 e dalla base del transistor T6 collegati in parallelo, che provoca la diminuzione esponenziale della VBE del transistor T7 fino a quando il suo valore è così basso che il transistor non conduce più e si interdice. Grazie alla resistenza di pull-up il piedino 3 del modulo amplificatore va a livello alto accendendo il PAM8302A. Viceversa togliendo la corrente al circuito spegnendo l'interruttore, il condensatore C2 si scarica rapidamente grazie alla resistenza R15 e il diodo D2, pronto per un nuovo ciclo di ricarica alla prossima accensione. Il segnale audio per i due amplificatori PAM8302A viene prelevato dalla presa jack da 3,5 mm della RPI e inviato al connettore CON6. I due condensatori C3 e C4 servono a disaccoppiare le alimentazioni dei due moduli PAM8302A, mentre il condensatore elettrolitico C1 serve per filtrare le fluttuazioni indesiderate di tensione e come riserva di corrente richiesta dagli amplificatori durante la riproduzione di brani ricchi di bassi. Il modulo RTC, basato sul chip DS3231, a catalogo codice 3085-DS3231RTC, viene collegato direttamente alla linea I²C del connettore d'espansione della RPI rispettivamente linea SDA pin 3 e linea SCL pin 5. Il modulo RTC viene fornito privo della batteria tampone CR2032 che dobbiamo acquistare separatamente e inserire nell'apposito portapile di forma circolare. Il varistore D3 serve a smorzare e proteggere la Raspberry Pi da eventuali picchi di tensione sulla linea a 5 V mentre il diodo D1 posto in parallelo alla bobina del relè RL1 serve a sopprimere le extra tensioni di verso contrario che si inducono allo spegnimento del relè.

SCHEMA ELETTRICO SENSORE HTU21D

Passiamo allo schema elettrico del sensore di temperatura e umidità HTU21D. Sulle linee di bus I²C dei due sensori sono presenti due resistenze di pull-up, una per la linea SCL e una per quella SDA, e servono per generare un potenziale alto visto che le uscite possono solo portare la linea a massa. Il valore dipende dalla capacità del cavo cioè dalla sua lunghezza, dal consumo cioè dal numero di dispositivi connessi e dalla velocità di trasmissione dati. Il valore minimo è dato dalla corrente massima che le uscite possono cortocircuitare a massa che si aggira attorno ai $1,5 \div 2$ Kohm. Poiché ogni sensore che inserite sul bus I²C è provvisto delle

resistenze di pull-up, la somma delle resistenze in parallelo di tutti i sensori per ogni linea non dovrebbe essere inferiore a questo valore. Se le linee sono corte e notate problemi di comunicazione sul BUS I²C, togliete le resistenze di pull-up da tutti i sensori lasciando solo quelle da 2,2 Kohm presenti sulla scheda principale del nostro MirrorCoppo, valore più che sufficiente per far funzionare il bus I²C a 400 kHz.

REALIZZAZIONE PRATICA

Finito di analizzare lo schema elettrico è giunta l'ora dell'assemblaggio. Per la realizzazione dei circuiti stampati per fotoincisione, potete partire dai file Gerber disponibili per il download sul nostro sito www.elettronica.in.it e ricavare da essi le pellicole mediante stampa in grandezza naturale su carta da lucido, su acetato oppure su carta oleata. Si avete capito bene proprio su carta oleata, anche se sembra strano si ottengono ottimi risultati anche con la carta oleata. A chi interessa approfondire l'argomento può visitare il link <https://www.youtube.com/watch?v=pOHEuExFzw> e guardare il video tutorial, realizzato dagli alunni dell'istituto Pietro Coppo, dove viene spiegato come realizzare un PCB mono faccia per fotoincisione di una basetta presensibilizzata, partendo appunto dal negativo del tracciato stampato su carta oleata. Come sempre consigliamo l'uso di uno stagnatore poco potente o meglio ancora una stazione saldante professionale con la regolazione della temperatura simile a quella disponibile a catalogo Futura Elettronica con il codice "VTSSC78". Per la saldatura dei componenti SMD invece consigliamo una stazione saldante ad aria calda simile a quella disponibile da Futura Elettronica con il codice "VTSS210" oppure quella abbinata ad un saldatore a stilo codice "VTSS230". Come stagno consigliamo sempre uno di ottima qualità, soprattutto per i componenti SMD, con diametro non superiore i 0,7 mm. Una volta ottenuto il PCB consigliamo di togliere il photoresist rimasto dallo sviluppo con un batuffolo di ovatta imbevuto di acetone e ricoprire le piste con un velo di stagno, per proteggere il rame dall'ossidazione, aumentare la resistenza meccanica e facilitare la successiva saldatura dei componenti. Dove necessario forate le piazzole per i componenti a foro passante con una punta da trapano da 0,8 mm oppure 1 mm per i morsetti a vite. Stagnate per primi sempre i componenti a profilo basso come i componenti SMD e successivamente quelli a profilo alto come quelli a foro passante. Fate attenzione al corretto posiziona-



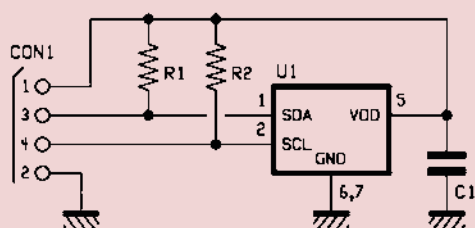
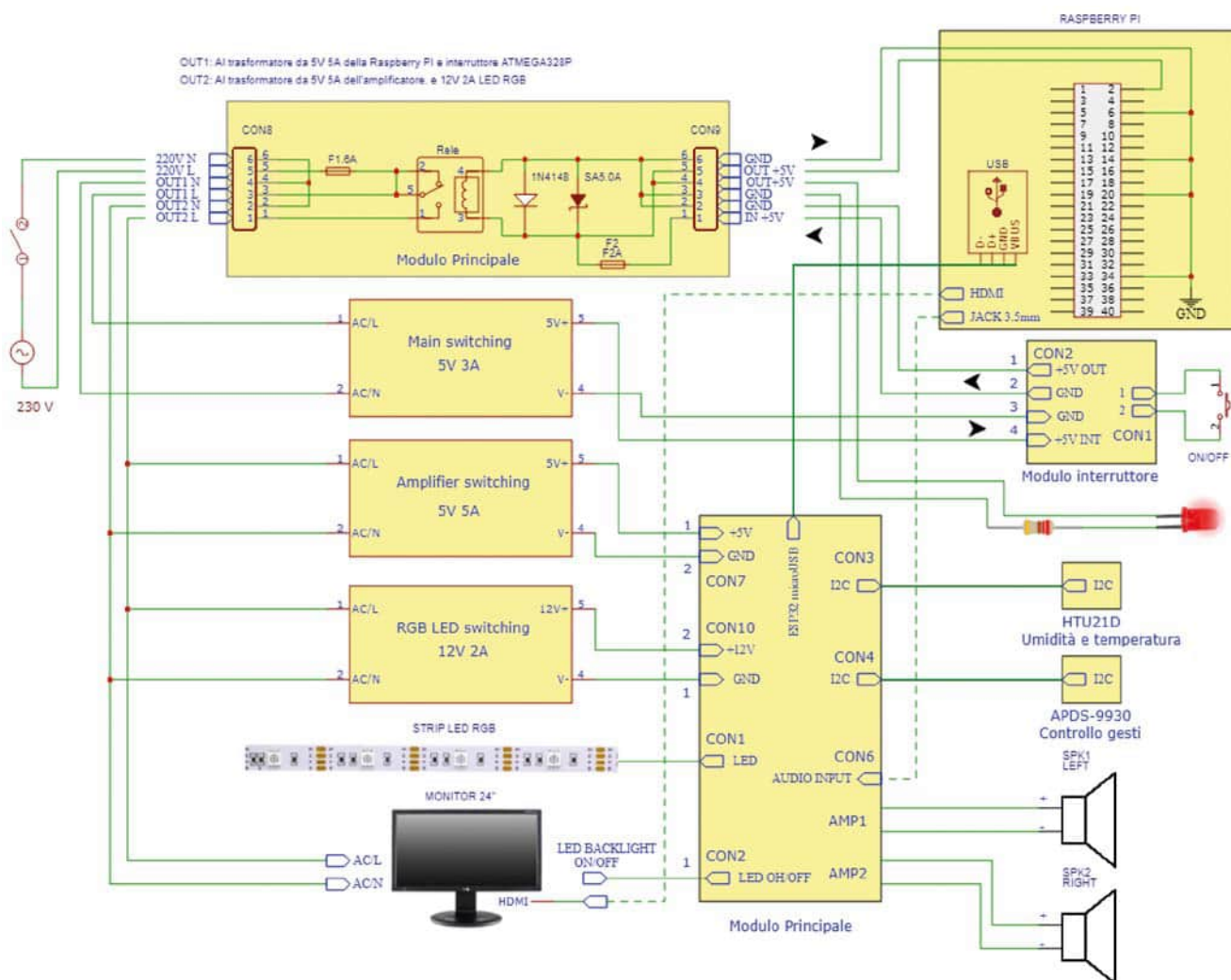


Fig. 5
Schema a
blocchi del
MirroCoppo.

mento rispettando il verso dei componenti attivi e quelli polarizzati. Durante il posizionamento dei componenti SMD utilizzate pinzette a punta sottile amagnetiche e utilizzate una lente di ingrandimento o meglio ancora un microscopio. Per i moduli ESP32, e il modulo RTC consigliamo di non saldarli direttamente al PCB ma di utilizzare strip maschio e femmina passo 2,54 mm, mentre per gli amplificatori PAM8302A consigliamo di saldarli

direttamente al PCB utilizzando uno strip maschio da 2,54 mm in questo modo non interferirà con il cavo micro USB per interfacciare il modulo ESP32 alla Raspberry Pi. Sugli amplificatori è presente un piccolo trimmer SMD che serve per la regolazione del volume in uscita. Per limitare la potenza sui diffusori, che sono piccoli e poco potenti, e di conseguenza anche la distorsione armonica totale (in inglese *total harmonic distortion*, da cui l'acronimo THD) consigliamo di posizionarli a metà corsa e successivamente, a progetto terminato, tararli ascoltando un brano musicale valutando la qualità sonora. Per chi è alle prime armi con la saldatura ad aria calda suggeriamo di fare attenzione nel saldare il microcontrollore ATMEGA328P sulla scheda interruttore. Cospargete il tracciato con il flussante e posizionate il componente facendo attenzione al corretto orientamento centrando i piedini alle



rispettive piazzole. Riscaldate con l'aria calda lo stampato attorno al microcontrollore con movimenti circolari e solo successivamente concentrate il flusso di aria calda su di esso fino alla completa stagnatura. Fate attenzione a non surriscaldarlo altrimenti potrebbe danneggiarsi. Usate questa tecnica per stagnare tutti i componenti SMD attivi e passivi.

CABLAGGIO

Terminato il montaggio dei circuiti stampati iniziamo a spiegare come cablare tra di loro i moduli e gli alimentatori. Per facilitare il cablaggio potete aiutarvi con lo schema a blocchi visibile in **Fig. 5**. La corrente elettrica a 220 V dell'impianto domestico entra sui piedini 5 e 6 del connettore CON8 passa attraverso il fusibile da 1,6 A e va ad alimentare l'alimentatore switching da 5 V 3 A. L'uscita va collegata ai piedini 3 e 4 del connettore CON2 della scheda interruttore facendo attenzione alla corretta polarità.

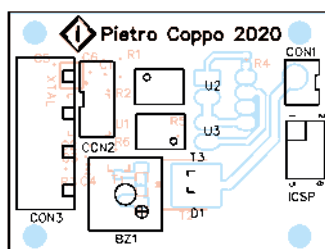
Sui piedini 1 e 2 del connettore CON1 collegherete un pulsante per il controllo dell'accensione del MirrorCoppo. Schiacciando il pulsante per 3 secondi, la corrente dai piedini 1 e 2 del connettore CON2 inizierà a scorrere verso i piedini 1 e 2 del connettore CON9. Si accenderà così la Raspberry Pi connessa ai piedini 5 e 6 e la spia LED, connessa ai piedini 3 e 4 del connettore CON9. Allo stesso tempo si attiva il relè accendendo i due alimentatori switching, rispettivamente quello da 5 V 5 A per alimentare

gli amplificatori di potenza in classe D e quello da 12 V 2 A per alimentare le strisce LED RGB. Il primo trasformatore lo collegherete al connettore CON7 e il secondo al connettore CON10. La striscia a LED verrà connessa al connettore CON1.

Il sensore temperatura e umidità HTU21D lo collegherete al connettore CON3 mentre il sensore del controllo gesti APDS-9960 lo collegherete al connettore CON4. Il connettore CON6 lo collegherete alla presa Jack da 3,5 mm della Raspberry Pi. La microUSB del modulo ESP32 la collegherete ad una delle prese USB libere della RPi, mentre l'uscita HDMI la collegherete alla presa HDMI del monitor.

Il miglior metodo per accendere e spegnere il monitor in modo istantaneo e quello di pilotare direttamente i driver della retroilluminazione LED, a seconda del modello, con un relè oppure un foto accoppiatore. Di solito al loro interno i monitor hanno due circuiti stampati quello dell'alimentatore e quello del controllo dell'immagine. I driver dell'accensione della retroilluminazione LED di solito si trovano sul circuito stampato dell'alimentazione. Consigliamo comunque di consultare lo schema elettrico del vostro monitor scaricandolo dal web. Chi non è capace di apportare questa modifica, dovrà accontentarsi di controllare l'accensione del monitor con un relè posto sulla linea di alimentazione a 220 V, e aspettare, ad ogni accensione, che scompaia lo splash screen (schermata di caricamento) con il logo della casa costruttrice.

scheda INTERRUPTORE



Elenco Componenti:

R1: 2,2 kohm
R2, R3, R6, R7: 10 kohm
R4: 120 ohm
R5: 220 ohm
C1, C2, C3: 100 nF ceramico
C4: 10 µF 5,6 V ceramico
C5, C6: 22 pF ceramico
D1: LL4148
T1, T3: 2N7002BKW,115
T2: IPB85P04P4L04
XTL1: 16 MHz
BZ1: KSSGJ4D20
U1: ATMEGA328P-MU
U2,U3: PC817C

CON1: Connettore JST 2 poli maschio
P. 2 mm
CON2 : Connettore JST 4 poli maschio
P. 2 mm
CON3: Morsetto 4 poli P. 5 mm
ICSP: Strip femmina 2x3 vie P. 2,54 mm

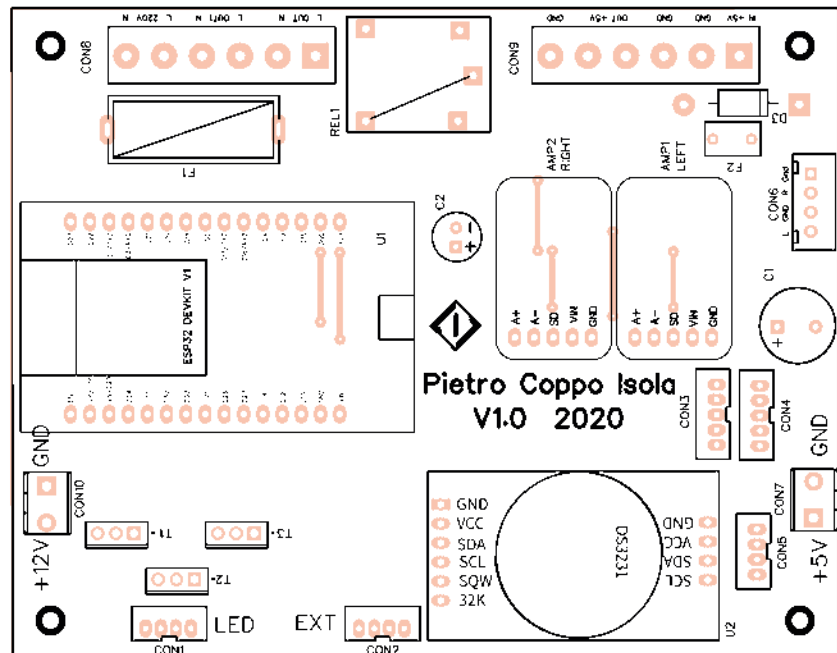
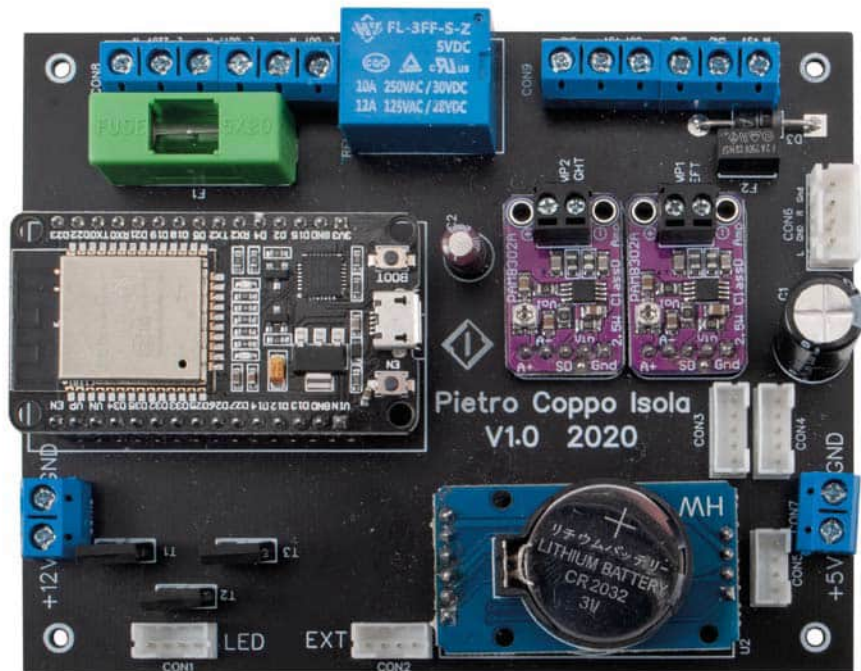
Varie
- Pulsante 6x6 mm
- Circuito stampato S1559_I (43x32,5 mm)

Elenco Componenti:

- R1, R3, R5: 4,7 kohm
- R2, R4, R6: 22 kohm
- R7, R9: 1,5 kohm
- R8, R10, R13, R14, R17, R18: 10 kohm
- R11, R12: 2,2 kohm
- R15: 47 kohm
- R16: 150 kohm
- JP1: vedi testo
- C1: 2200 μ F 6,3 VL elettrolitico
- C2: 4,7 μ F 63 VL elettrolitico
- C3, C4: 100 nF 16 V ceramico
- T1, T2, T3: BD679A
- T4, T5, T6, T7: MMBT3904
- D1, D2: 1N4148
- D3: SA5.0A
- U1: Modulo ESP32
- U2: Modulo DS3231
- AMP1, AMP2: Modulo PAM8302A
2,5 W mono
- F1: Fusibile 1,6 A 5x20 mm
- F2: Fusibile 2 A passo 5,1 mm
- CON1, CON2, CON5: Connettore JST
4 poli maschio P. 2 mm
- CON3, CON4: Connettore JST 5 poli
maschio P. 2 mm
- CON6: Connettore JST 4 poli maschio
P. 2,54 mm
- CON7, CON10: Morsetto 2 poli P. 5 mm
- CON8, CON9: Morsetto 6 poli P. 5 mm
- REL1: Relé 5 V 10 A 1 scambio

Varie

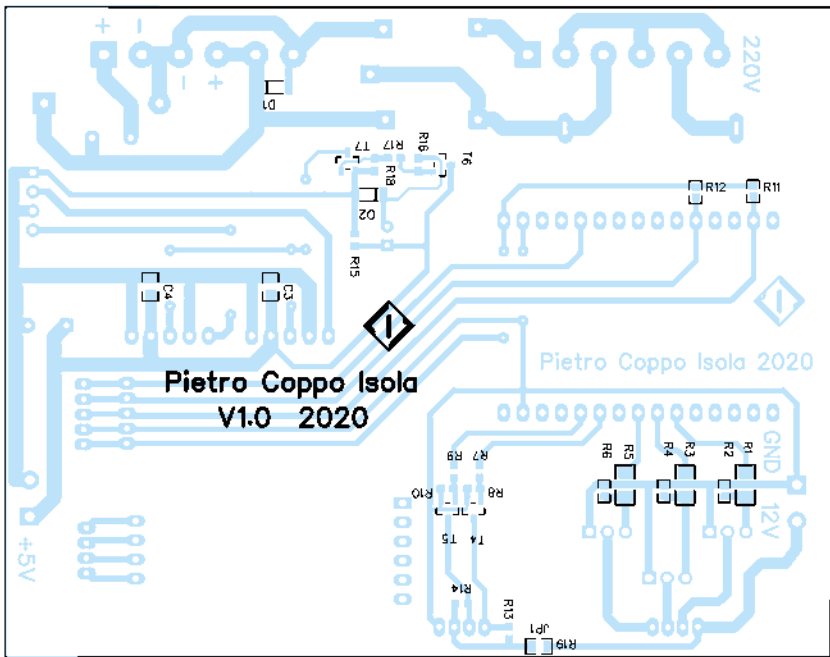
- Strip femmina 4 vie P. 2,54 mm
- Strip femmina 6 vie P. 2,54 mm
- Strip femmina 15 vie P. 2,54 mm
- Strip maschio 6 vie P. 2,54 mm
- Strip maschio 4 vie P. 2,54 mm
- Spina Jack 3,5 mm stereo
- Porta fusibile L 22,6 - W 9 mm
- Altoparlante 8 ohm 3 W
- Circuito stampato S1559_P
(86,5x110 mm)



PROGRAMMAZIONE

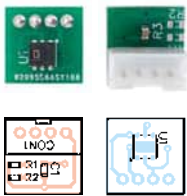
Prima di montare il modulo ESP32 sul circuito stampato della scheda principale bisogna programmarlo compilando il sorgente *MirrorCoppo.ino* mediante l'ambiente di sviluppo Arduino IDE, che potete scaricare dal sito web www.arduino.cc.

Facciamo presente che per programmare, dall'ambiente Arduino Ide, i moduli dell'Espressivo ESP32, ESP8266 oppure ESP8285 bisogna scaricare da Internet alcuni pacchetti aggiuntivi con le configurazioni delle suddette schede. In *Arduino Ide* dal menu "File" selezionate il sottomenu "Impostazioni"



- http://arduino.esp8266.com/stable/package_esp8266com_index.json;
- https://dl.espressif.com/dl/package_esp32_index.json".

Elenco Componenti:



R1, R2: 10 kohm
C1: 100 nF ceramico
CON1: Connettore JST 4 poli maschio P. 2 mm
U1: HTU21DSMD

Varie

- Circuito stampato S1559_S (10,5x11 mm)

Fig. 6
Pagina del captive
portal dell'ESP32.



Vi serviranno alcune librerie aggiuntive che dovete scaricare dal menu "Sketch", sottomenu "#include librerie" selezionando "Gestione librerie", da qui installerete le librerie `iotWebConf`, `Adafruit_HTU21DF`, `SparkFun_APDS9960`, `WebSocketsClient` e `ArduinoJson`. A causa di incompatibilità con le librerie `ArduinoJson` versione 6, durante l'installazione dal menu a tendina scegliete la più

Fig. 7
Pagina di
configurazione
del'ESP32.



recente tra quelle della versione 5, cioè la versione 5.13.5. Ricordiamo che il modulo ESP32 acquistato presso il negozio online www.futurashop.it, codice prodotto YB555, prima di programmarlo bisogna metterlo in modalità programmazione, premendo prima il pulsante destro, quello contrassegnato con la sigla *BOOT*, e successivamente quello sinistro del *reset*, contrassegnato con la sigla *EN*, per poi rilasciarli entrambe, altrimenti non riuscirete a programmarlo. Fate attenzione a non schiacciare contemporaneamente i due pulsanti *BOOT* ed *EN* per un periodo troppo lungo, altrimenti il modulo va in modalità programmazione firmware e potreste rischiare di bloccarlo. Un metodo alternativo è premere il tasto *BOOT* mentre si inserisce il cavo di programmazione nella presa USB, così il modulo va automaticamente in modalità programmazione senza rischio di sbagliare. Una volta programmato il modulo ESP32 all'accensione si comporterà come un access point (punto d'accesso) captive portal. Questo significa che potete connettervi al modulo ESP32, come fate con la rete WiFi del router di casa vostra, questo per consentire l'accesso al pannello di configurazione. Questa modalità si attiva ad ogni accensione per un tempo impostabile (*startup delay*) dal pannello di configurazione, oppure quando l'ESP32 non è configurato correttamente o non è in grado di collegarsi ad una rete wireless. Fate una scansione delle reti wireless con il vostro PC, selezionate la rete "ESP32-MirrorCoppo" e alla richiesta inserite la password predefinita "123456789". In automatico si avvierà il browser predefinito del sistema operativo, visualizzando la pagina simile a quella di **Fig. 6**. Selezionando il pulsante "Configuration" si aprirà una nuova pagina simile a quella di **Fig. 7**, dove potete modificare la password, inserire il nome SSID e la password della rete WiFi domestica.

Come spiegato precedentemente il campo *startup delay* vi permette di impostare la durata in secondi di attivazione del captive portal all'accensione del dispositivo. Prima di compilare i rimanenti campi, consigliamo di salvare la configurazione e verificare che l'ESP32 acceda correttamente alla vostra rete WiFi domestica. Per conoscere l'indirizzo IP della rete locale assegnato all'ESP32, potete leggerlo nella sezione delle connessioni di rete del vostro router oppure attivando la modalità debug *DBSER* nel sorgente, prima della compilazione del sorgente, così il numero IP verrà visualizzato sul terminale assieme ad altre informazioni utili. Per non avere problemi di comunicazione con la RPI usate la modalità debug esclusivamente in fase di

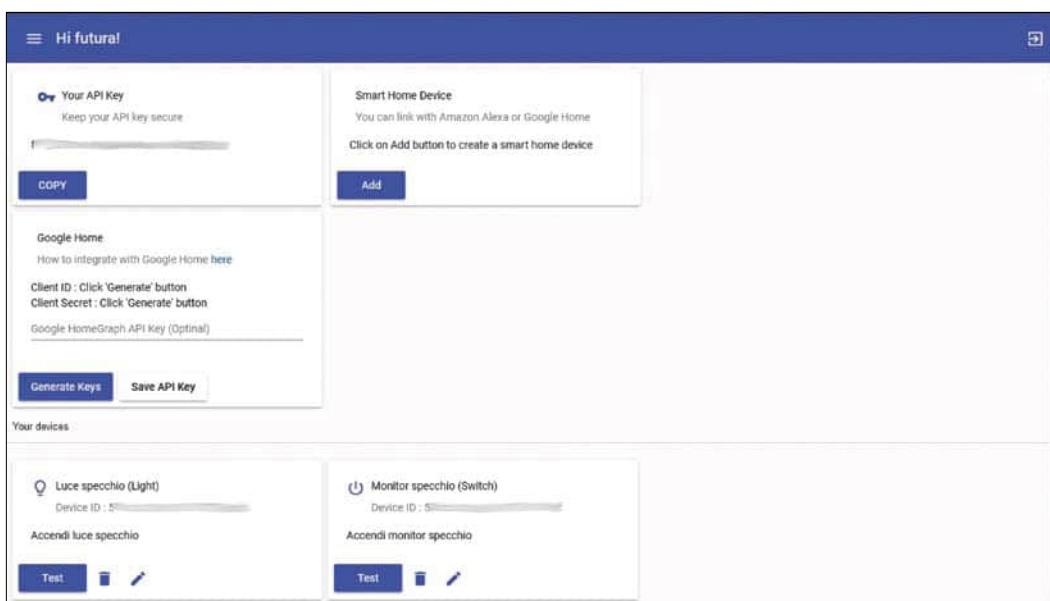


Fig. 8
Pagina iniziale del sito Sinric, con con i due device “Luce Specchio” e “Monitor specchio”.

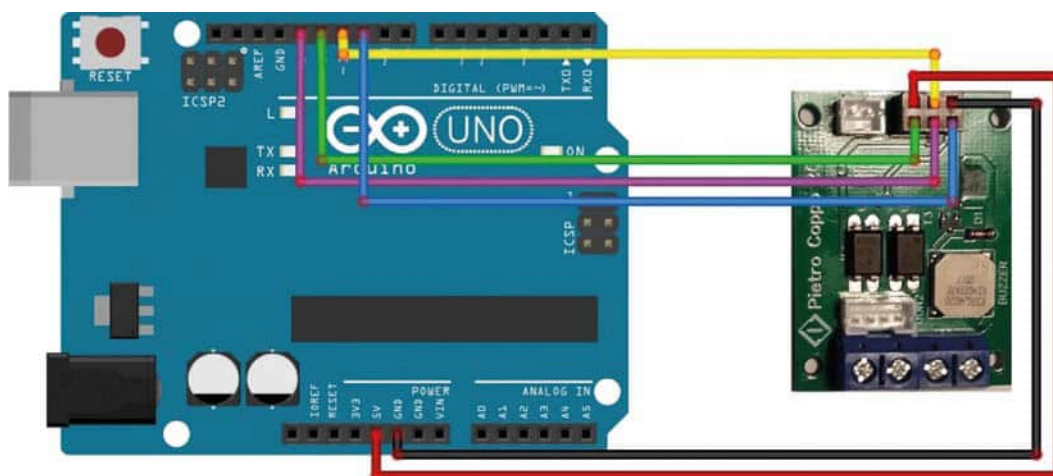
progettazione, quando andate ad apportare modifiche al sorgente, e disattivatela successivamente commentando la riga *DBSER*. Per creare le chiavi da inserire nei campi del pannello di configurazione dovete registrarvi al sito <http://www.sinric.com> e una volta effettuato l'accesso nella schermata principale aggiungete due nuovi dispositivi *Luce specchio* e *Monitor specchio* come visibile in **Fig. 8**. Per aggiungere un nuovo dispositivo cliccate sul tasto *Add* in *Smart Home Device*, vi si aprirà una nuova finestra e nel campo *Friendly name* scrivete “*Luce specchio*”, in *description* scrivete “*Accendi luce specchio*” e dal menu a tendina *type* scegliete “*Light*”. Per aggiungere il secondo dispositivo cliccate di nuovo su *Add* nel campo *Friendly name* scrivete “*Monitor specchio*”, in *description* scrivete “*Accendi monitor specchio*” e dal menu a tendina *type* scegliete “*Switch*”. *Friendly name* è il nome del dispositivo, mentre *type* definisce il tipo di dispositivo che Alexa dovrà controllare. Definendo il dispositivo come *light*, Alexa lo gestisce come una lampadina di tipo RGB che può cambiare colore e può variare l'intensità luminosa. Viceversa scegliendo il dispositivo come *switch*, Alexa lo considera come se fosse un normale interruttore e vi permetterà solo di accenderlo o spegnerlo. Una volta creati i dispositivi, copiate l'*Api Key*, il *Device ID* di “*Luce specchio*” e il *Device ID* di “*Monitor specchio*” inserendoli nei rispettivi campi nella pagina di configurazione dell'ESP32, mentre nel campo *server* inserite il nome del server “*iot.sinric.com*”. Vi consigliamo di copiare i codici usando gli appositi pulsanti, copiando le chiavi manualmente si rischia

di copiare involontariamente anche lo spazio all'inizio o alla fine della chiave, che inevitabilmente provocherà un errore di connessione. Terminata la programmazione e la configurazione dell'ESP32 potete montarlo sul PCB facendo attenzione affinché i piedini si inseriscano correttamente nei fori dello strip.

Passiamo ora alla programmazione dell'ATMEGA328P della scheda interruttore. Poiché il microcontrollore è lo stesso della scheda Arduino Uno potete programmarlo utilizzando l'ambiente di sviluppo Arduino IDE tramite il connettore ICSP. Come programmatore potete usare una scheda Arduino Uno programmata come *ISP programmer* collegandola alla scheda interruttore secondo lo schema di **Fig. 9** e consultando la **Fig.10** e la **Tabella 2** per le connessioni al connettore ICSP. Poi, dal menu strumenti selezionate la scheda *Arduino Pro or Pro mini* e come processore selezionate *Atmega328P(5v, 16MHz)*. Dal sottomenu “*programmatore*” del menu a tendina “*strumenti*” selezionate la voce “*Arduino as ISP*”, selezionate la porta seriale corretta e infine selezionate “*Scrivi il bootloader*”. Terminato il caricamento del bootloader potete programmare il microcontrollore compilando lo sketch *interruttore.ino* che potete scaricare dal sito web www.futuraelettronica.it.

A questo proposito se utilizzate come programmatore ISP un clone cinese della scheda Arduino Uno e riscontrate problemi dopo la programmazione dell'ATMEGA328P della scheda interruttore, come il suono distorto e poco squillante del buzzer alla pressione del pulsante, consigliamo di aggiornare

Fig. 9
Schema di collegamento tra la scheda Arduino Uno, usata come programmatore ISP, e il connettore ICSP della scheda interruttore.



il bootloader della scheda Arduino Uno caricandolo dall'ambiente di sviluppo Arduino IDE. Ricordiamo che la scheda interruttore potete utilizzarla in qualsiasi progetto per controllare in modo sicuro l'accensione e lo spegnimento della RPI, facendo fare al sistema operativo spegnimenti puliti senza danneggiarlo. Ovviamente editando adeguatamente il file di configurazione della RPI `/boot/config.txt` come già spiegato nell'articolo.

FUNZIONAMENTO

L'assistente vocale Amazon Alexa può essere controllato sia con richieste vocali, sia con l'app Amazon Alexa installata sullo smartphone. Esistono innumerevoli applicazioni chiamate skill che si possono attivare solo mediante l'app Amazon Alexa. Quindi se volete ascoltare la vostra radio preferita con il MirrorCoppo ricordate che dovette prima abilitare la skill associata alla radio tramite l'app installata sullo smartphone. Di seguito riportiamo alcuni esempi di richieste vocali:

Di seguito riportiamo alcuni esempi di richieste vocali:

- "Alexa, accendi luce specchio."
- "Alexa, spegni luce specchio."
- "Alexa, intensità luce 30%."
- "Alexa, colore specchio blu."
- "Alexa, incrementa luce specchio."
- "Alexa, decrementa luce specchio."
- "Alexa, accendi monitor specchio."
- "Alexa, spegni monitor specchio."
- "Alexa, volume quattro."
- "Alexa, metti della musica."

Per conoscere tutte le funzionalità e il nome corretto di tutti i colori e le tonalità riconosciute, facciamo presente che ci sono ben cinque tonalità di luce bianca, vi consigliamo di andare a consultare la sezione dispositivi dell'app Amazon Alexa.

Fig. 10
Piedinatura del connettore ICSP AVR.

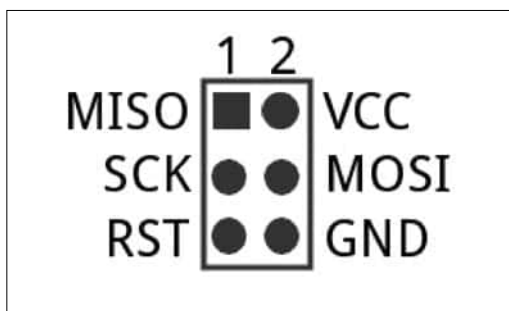


Tabella 2
Collegamenti tra la scheda Arduino Uno e il connettore ICSP per caricare il bootloader sulla scheda interruttore.

ARDUINO (PROGRAMMATORE ICSP)	SCHEDA INTERRUETTORE
Arduino Pin 12 – MISO	ICSP Pin 1 – MISO
Arduino – VCC +5V	ICSP Pin 2 – VCC
Arduino Pin 13 – SCK	ICSP Pin 3 – SCK
Arduino Pin 11 – MOSI	ICSP Pin 4 – MOSI
Arduino Pin 10 – Reset	ICSP Pin 5 – Reset
Arduino – GND	ICSP Pin 6 – GND

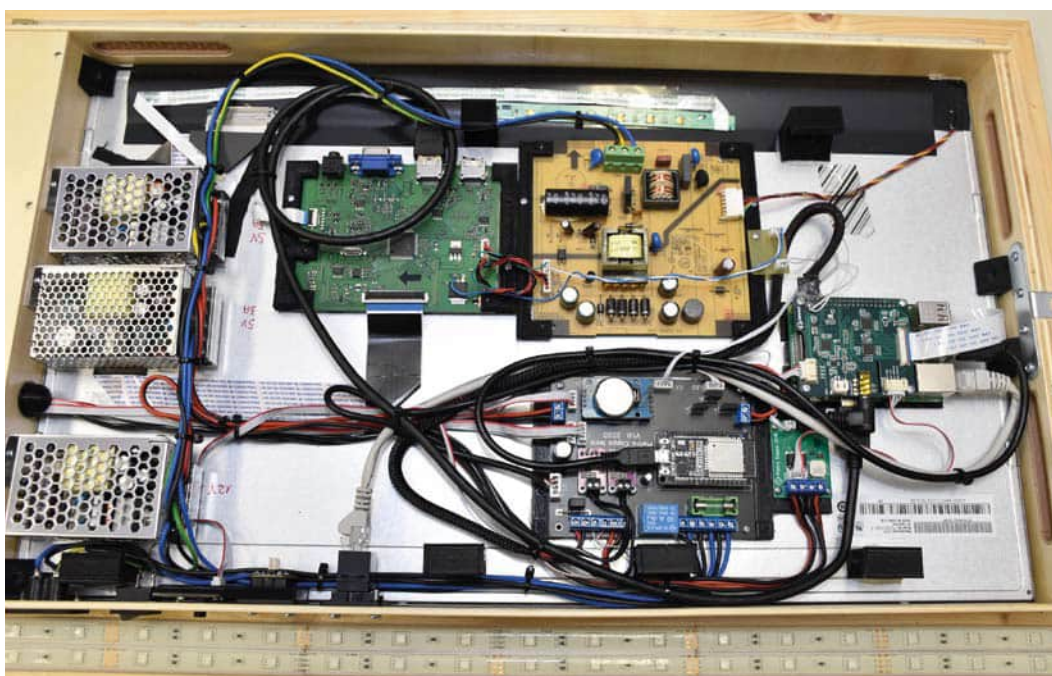


Fig. 11
Assemblaggio
dello
specchio
MirroCoppo
visto da
dietro.

Per aggiungere i dispositivi dall'app Amazon Alexa, basta premere il pallino bianco con il simbolo più in alto a destra e selezionare "Aggiungi dispositivo" scorrere verso il basso selezionando "Altro" e confermare con il tasto "Trova dispositivi". I dispositivi trovati, potete raggrupparli in gruppi a seconda della stanza in cui si trovano. Così facendo potete gestirli tutti assieme con più facilità e con un solo comando accenderli o spegnerli contemporaneamente. Poiché questa tecnologia è basata sul cloud è necessaria una connessione Internet attiva per poterla utilizzare, abbiamo pensato così di aggiungere un dispositivo di controllo a gesti quando la connessione Internet viene a mancare, utilizzando il sensore APDS-9960. Sul nostro prototipo questo sensore è stato nascosto all'interno del fregio posizionato anteriormente nella parte bassa della cornice. Muovendo il palmo della mano davanti al sensore da sinistra a destra si accende la strip LED, mentre spostando la mano da sinistra a destra si spegne. Muovendolo dall'alto verso il basso il monitor si accende, viceversa muovendo la mano dal basso verso l'alto il monitor si spegne. Per accendere contemporaneamente sia il monitor che la strip LED basta avvicinare il palmo della mano al sensore, mentre allontanandola velocemente si spengono entrambe. Chi vuole vedere come funziona MirrorCoppo può visitare il canale YouTube della sezione informatici dell'istituto Pietro Coppo all'indirizzo https://youtu.be/c7fL-3paO_HU.

CONCLUSIONI

Spesso e volentieri le micro SD utilizzate con la Raspberry Pi si cancellano o si danneggiano irrimediabilmente a causa di sbalzi di tensione o in seguito a blackout improvvisi; perciò vi consigliamo di fare sempre una copia di backup della microSD con il programma Win32DiskImager e se possibile equipaggiare la Raspberry Pi con un gruppo di continuità come il PowerPi pubblicato sul n° 250. Con questo progetto speriamo di aver esaudito il desiderio di chi vuole realizzare il proprio assistente vocale o un sistema domotico avanzato, partendo magari da uno specchio smart che dà la sveglia al mattino e regala il buon umore con la musica preferita mentre ci si prepara ad affrontare una giornata lavorativa impegnativa. □



Cosa occorre?

I componenti principali utilizzati in questo progetto sono disponibili presso Futura Elettronica.

Il materiale va richiesto a:

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

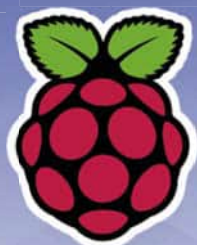
RASPBERRY PI 400!

La tastiera con il computer integrato

€ 146,00

Cod. PI400KIT-ITA

IVA inclusa



**Manuale
in italiano!**

Prezzo IVA inclusa.

Kit completo con computer Raspberry Pi 400 accessori e manuale in Italiano.



Raspberry Pi 400 è un personal computer completo che integra nella tastiera (italiana), una scheda Raspberry Pi 4 da 4Gb con processore quad-core a 64 bit, Wi-Fi, Bluetooth, doppia uscita video (micro HDMI) con una risoluzione fino a 4K, Gigabit Ethernet, USB 2.0 e USB 3.0.

Leggero, portatile, semplice da usare. È ideale per navigare sul web, creare e modificare documenti, guardare video e imparare a programmare utilizzando l'ambiente desktop del sistema operativo Raspberry Pi. I pin GPIO di Raspberry Pi rimangono comunque accessibili, permettendo di connettere componenti e prototipare progetti.

La confezione di Raspberry Pi 400 contiene tutto il necessario (ad eccezione di una TV o di un monitor): computer Raspberry Pi P400, alimentatore da rete, mouse, scheda microSD da 16GB pre-programmata con Raspberry Pi OS, cavo da Micro HDMI a HDMI A lungo 1 metro, un manuale illustrato a colori di 247 pagine completamente in Italiano.

RFID CON RC522



del Prof.
MICHELE MENNITI

Conosciamo e impariamo a utilizzare una scheda per l'identificazione di tag a RF, facilmente abbinabile ad Arduino.

A

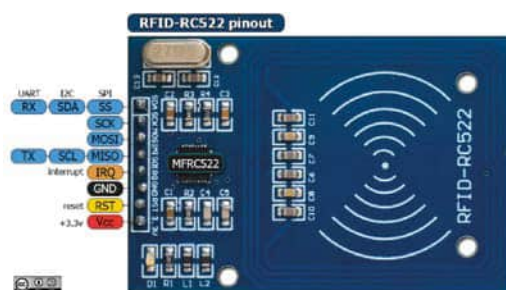
bbiamo più volte trattato il tema dell'identificazione a radiofrequenza, meglio nota come RFID; ad esempio nel fascicolo n° 254 le abbiamo dedicato un approfondimento attraverso una doviziosa introduzione teorica e alcuni esempi di utilizzo basati sul modulo RFID PN522, accompagnato dall'im-

mancabile scheda Arduino. In questo articolo riprendiamo il discorso sperimentando un altro modulo, l'RFID-RC522 basato sull'omonimo chip MFRC522 prodotto dalla NXP (il datasheet è reperibile al link web <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>).

L'MFRC522 è un lettore/scrittore RFID altamente integrato per comunicazioni senza contatto a 13,56 MHz, che supporta gli standard ISO/IEC14443A/MIFARE e NTAG.

Il trasmettitore interno all'MFRC522 è in grado di pilotare un'antenna progettata per comunicare con schede e tran-

Fig. 1
Il modulo
RFID-RC522.



sponder senza circuiti attivi aggiuntivi. Il ricevitore interno riesce in modo efficiente a demodulare e decodificare segnali provenienti da schede e transponder compatibili con gli standard suelencati. L'MFRC522 supporta i modelli MF1xxS20, MF1xxS70 e MF1xxS50. Supporta inoltre la comunicazione senza contatto, con velocità di trasferimento MIFARE fino a 848 kB (kiloBaud) in modalità bidirezionale.

L'MFRC522 è compatibile con tutte le varianti dei protocolli di identificazione MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 e MIFARE Plus RF, raggruppati sotto il termine generico di MIFARE.

Qui di seguito riepiloghiamo le caratteristiche tecniche principali del modulo:

- circuito analogico per demodulare e decodificare le risposte;
- driver bufferizzati per il collegamento di un'antenna con un numero ridotto di componenti esterni;
- supporta ISO/IEC 14443 A/MIFARE e NTAG;
- distanza operativa tipica in modalità di lettura/scrittura fino a 50 mm a seconda delle dimensioni dell'antenna e della sintonizzazione;

Tabella. 1
Connessioni fra il modulo RFID-RC522 e Arduino.

ARDUINO UNO	RFID-RC522	
3.3V	Rosso	3.3V
pin 9	Rosa	RST
GND	Nero	GND
pin 12	Verde	MISO
pin 11	Giallo	MOSI
pin 13	Arancione	SCK
pin 10	Blu	SS (SDA)

- supporta la crittografia MF1xxS20, MF1xxS70 e MF1xxS50 in modalità di lettura / scrittura;
- supporta ISO/IEC 14443 con comunicazione a velocità di trasferimento elevata fino a 848 kbaud;
- supporta MFIN/MFOUT;
- alimentazione interna aggiuntiva all'IC della smart card collegata tramite MFIN/MFOUT
- interfacce host supportate: SPI fino a 10 Mbit/s., I²C-Bus fino a 400 kBd in modalità Fast, fino a 3400 kBd in modalità High-speed, UART seriale RS232 fino a 1.228,8 kB/s, con livelli di tensione dipendenti dalla tensione di alimentazione dei pin;
- il buffer FIFO gestisce l'invio e la ricezione a 64 byte;
- modalità interrupt flessibili;
- hard reset con funzione di bassa potenza;
- spegnimento da software;
- timer programmabile;
- tensione di alimentazione da 2,5 a 3,3 V;
- Pin I/O programmabili.

Il modulo RFID-RC522 è dettagliato in **Fig. 1**, dove potete vedere tutte le connessioni disponibili. Poiché al momento tutte le librerie in circolazione per l'utilizzo del modulo con Arduino riutilizzano la connessione SPI, ci concentreremo solo su di essa, la **Fig. 2** illustra appunto il collegamento da effettuare per le successive prove con Arduino UNO. Nella **Tabella 1** descriviamo analiticamente i collegamenti con i relativi segnali: a sinistra quelli di Arduino UNO e a destra i contatti del modulo RFID-RC522 con i colori dei fili corrispondenti nello schema della **Fig. 2**.

Il pin IRQ del modulo RFID va lasciato scollegato perché in questo progetto non lo gestiamo e non ci occorre.

Naturalmente occorre fare molta attenzione, perché l'inversione di due soli cavetti tra loro comporterà il mancato funzionamento del modulo; altra cosa importante è l'alimentazione, il modulo richiede 3,3 V, il corrispondente pin su Arduino si trova proprio accanto a quello dei 5 V, una distrazione comporterebbe quasi certamente il guasto del modulo.

La scheda (il modulo RFID utilizzato nel progetto) è reperibile sul sito www.futurashop.it (2846-VMA405N) e viene fornita con due file strip da 8 pin (una dritta e l'altra a 90°) da saldare a scelta negli appositi fori dei segnali, un Tag Card RFID ed un Tag portachiavi RFID per poter fare le prime prove.

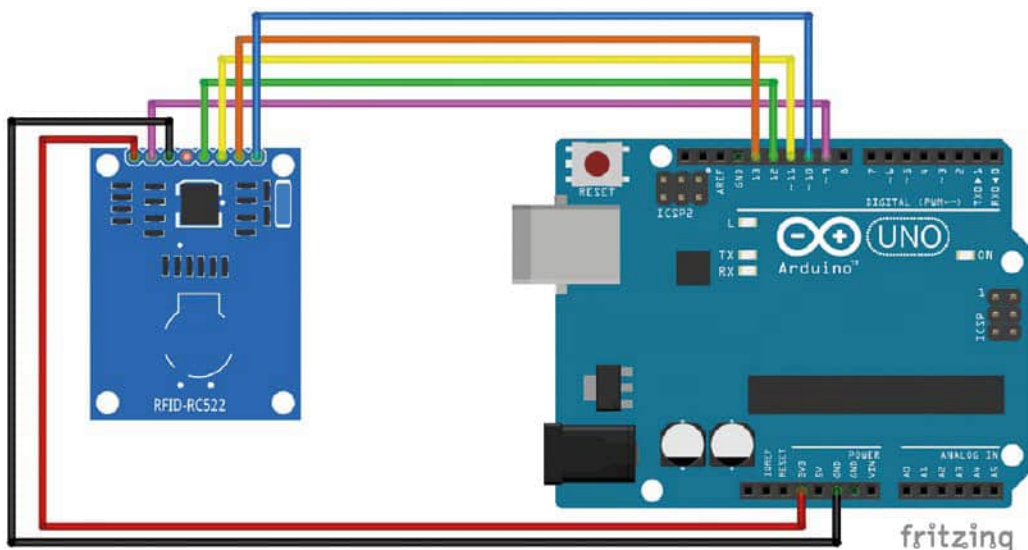


Fig.2
Collegamenti
RC522_ISP.

L'uso di questo modulo è facilitato dalla disponibilità di specifiche librerie, delle quali ne abbiamo selezionate due: quella della Velleman e quella scritta da Miguel Balboa.

Velleman: è una libreria minimal costituita da due soli file (RFID.h e RFID.cpp), una volta scaricata dal link: https://www.velleman.eu/downloads/29/vma405_library.zip occorre includerla aprendo un nuovo sketch con una versione aggiornata dell'IDE (nel nostro caso 1.8.12). Eseguire il comando **Sketch - #include libreria – Aggiungi libreria da file .ZIP...** e selezionare il file .zip (**Fig. 3**), quindi chiudere l'IDE; alla riapertura sarà disponibile. La libreria è scaricabile anche dalla sezione download del sito www.elettronica.in col nome "vma405_library.zip".

Miguel Balboa: questa libreria, molto più potente ed articolata, è consigliata dal sito ufficiale di Ar-

duino, viene costantemente aggiornata quindi conviene sempre andare sul link <https://www.arduino.cc/reference/en/libraries/mfrc522/> o direttamente sul github <https://github.com/miguelbalboa/rfid>; nel momento in cui stiamo scrivendo queste pagine l'ultima versione è la 1.4.8, che troverete sempre nella sezione download del sito www.elettronica.in, col nome "MFRC522-1.4.8.zip". La sua inclusione nell'IDE si ottiene con la stessa modalità vista prima. Questa libreria è piuttosto impegnativa e richiede notevoli competenze in materia RFID e di programmazione per poterla sfruttare, quindi ci limitiamo a segnalare ai lettori più esperti, mentre per il test che vedremo più avanti useremo quella distribuita dalla Velleman.

TEST DI FUNZIONAMENTO

Iniziamo con un test per verificare i collegamenti

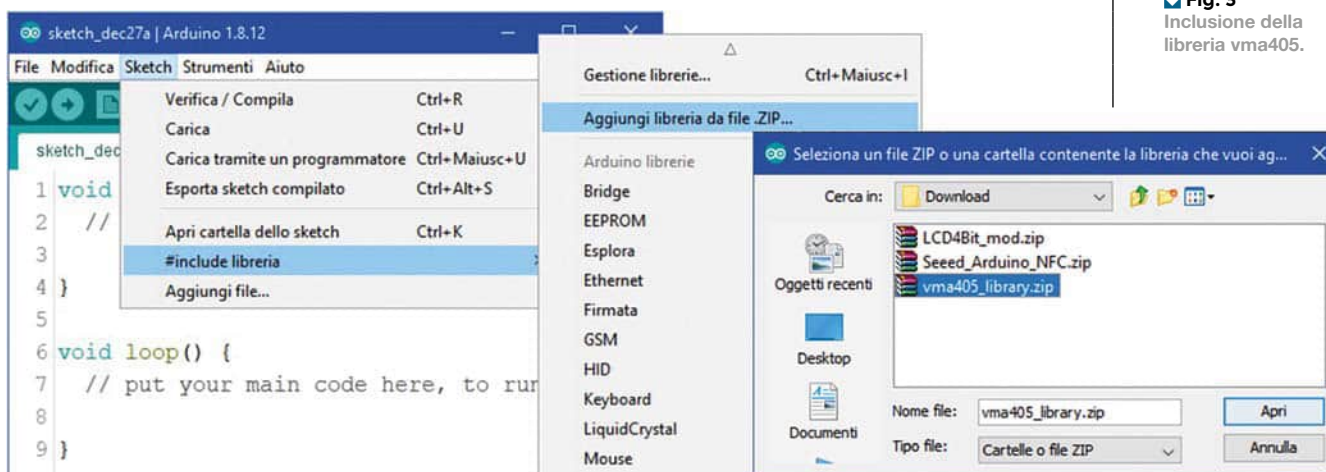


Fig. 3
Inclusione della
libreria vma405.

→ **Fig. 4**
Lettura di un Tag
col modulo RC522.



e la corretta installazione della libreria *vma405*. Nella sezione download del sito *www.elettronicaain.it* abbiamo inserito uno sketch denominato "RFID_TAG_CODE", occorre aprirlo nell'IDE e caricarlo su Arduino UNO, quindi bisogna aprire il Monitor Seriale che mostrerà il messaggio "**Avvicina un Tag al Modulo**"; a questo punto avviciniamo il Tag alla parte del modulo su cui sono serigrafati i simboli delle onde radio (**Fig. 4**).

Bisogna tener presente che il modulo va sempre tenuto distante da dispositivi in grado di generare campi elettrici o elettromagnetici, altrimenti potrebbe non funzionare correttamente.

Nell'istante in cui il Tag viene riconosciuto, il modulo ne legge l'UID (codice identificativo univoco) e lo visualizza sul Monitor Seriale, sia in formato decimale che in formato esadecimale; in **Fig. 5** possiamo osservare la lettura successiva di due Tag.

Se dopo la prima identificazione si avvicina nuovamente lo stesso Tag, non succederà nulla in quanto nello sketch non abbiamo previsto un'azione alter-

nativa, avvicinando invece un altro Tag si otterrà il relativo UID nei due formati.

Nel momento in cui questo test va a buon fine, occorre decidere quale Tag sarà l'"Amministratore di sistema" del circuito che abbiamo preparato per la sperimentazione di questo modulo e prendere nota dei cinque numeri decimali che compongono il suo UID.

Per il nostro esempio useremo il primo gruppo visibile nella figura (197, 5, 190, 35, 93).

PROGETTO CONTROLLO ACCESSI

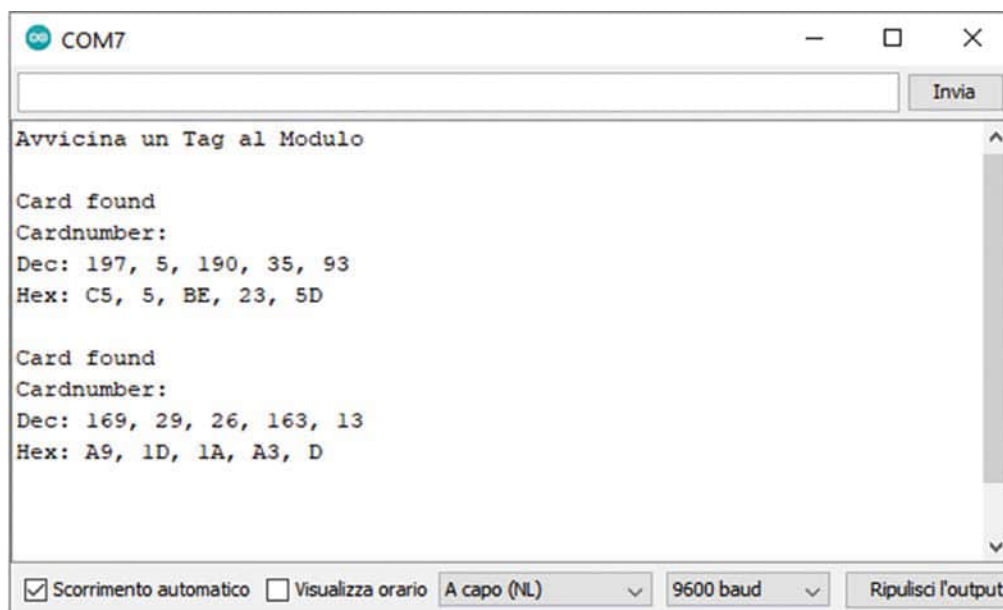
Con l'ausilio di una breadboard e pochissimi altri componenti, passiamo ora alla realizzazione di un circuito per simulare un controllo di accessi mediante tecnologia RFID impiegando allo scopo il nostro modulo RC522.

Lo sketch che abbiamo predisposto si può scaricare dalla sezione download del sito *www.elettronicaain.it* e si chiama *Access_Control.rar*, è composto da due schede e la cartella ricavata dalla decompressione del file dovrà essere posizionata nella cartella degli sketch dell'IDE in vostro possesso; in precedenza dovrà essere stata installata la libreria **vma405_library**.

L'idea è quella di un ipotetico controllo di accesso, in cui l'accettazione del Tag viene indicata con l'accensione/spegnimento alternati di un LED blu, mentre il rifiuto è rappresentato dall'accensione di un LED rosso per circa 3 secondi.

I Tag autorizzati vengono pre-memorizzati nella EEPROM del microcontrollore di Arduino; il nostro

→ **Fig. 5**
Lettura dell'UID
di due Tag.



sketch ne prevede massimo tre ma possono essere aumentati fino alla disponibilità della EEPROM. La memorizzazione avviene solo previa autorizzazione dell'“Amministratore di sistema” che possiede un Tag CARD RFID, il cui UID deve essere inserito manualmente nello sketch e che, una volta letto dal modulo, permette di avvicinare altri Tag per accreditarli e memorizzarli nella EEPROM. Da questo punto in poi avvicinando al modulo un Tag autorizzato il LED blu si accenderà o si spegnerà ad ogni passaggio, invertendo quindi il suo stato. Il LED rosso invece si accenderà per tre secondi quando si proverà ad accedere mediante un Tag non registrato, negando di fatto l'autorizzazione, quindi si spegnerà. Prima di caricare lo sketch nell'Arduino occorre quindi aprirlo ed assegnare alle cinque variabili

```

12 //Codice della chiave amministratore
13 #define dec_code0 197
14 #define dec_code1 5
15 #define dec_code2 190
16 #define dec_code3 35
17 #define dec_code4 93

```

Fig. 6
Inserimento dell'UID amministratore nello sketch.

dec_code0 ÷ dec_code4 i corrispondenti valori decimali dell'UID del Tag prescelto (**Fig. 6**), inserendoli al posto di quelli che troverete pre-memorizzati. A questo punto si può caricare lo sketch nell'Arduino e successivamente scollegare il cavo USB per togliere l'alimentazione, quindi si passa alla realizzazione dello schema pratico visibile in **Fig. 7**, che vi spiega come collegare gli elementi occorrenti.

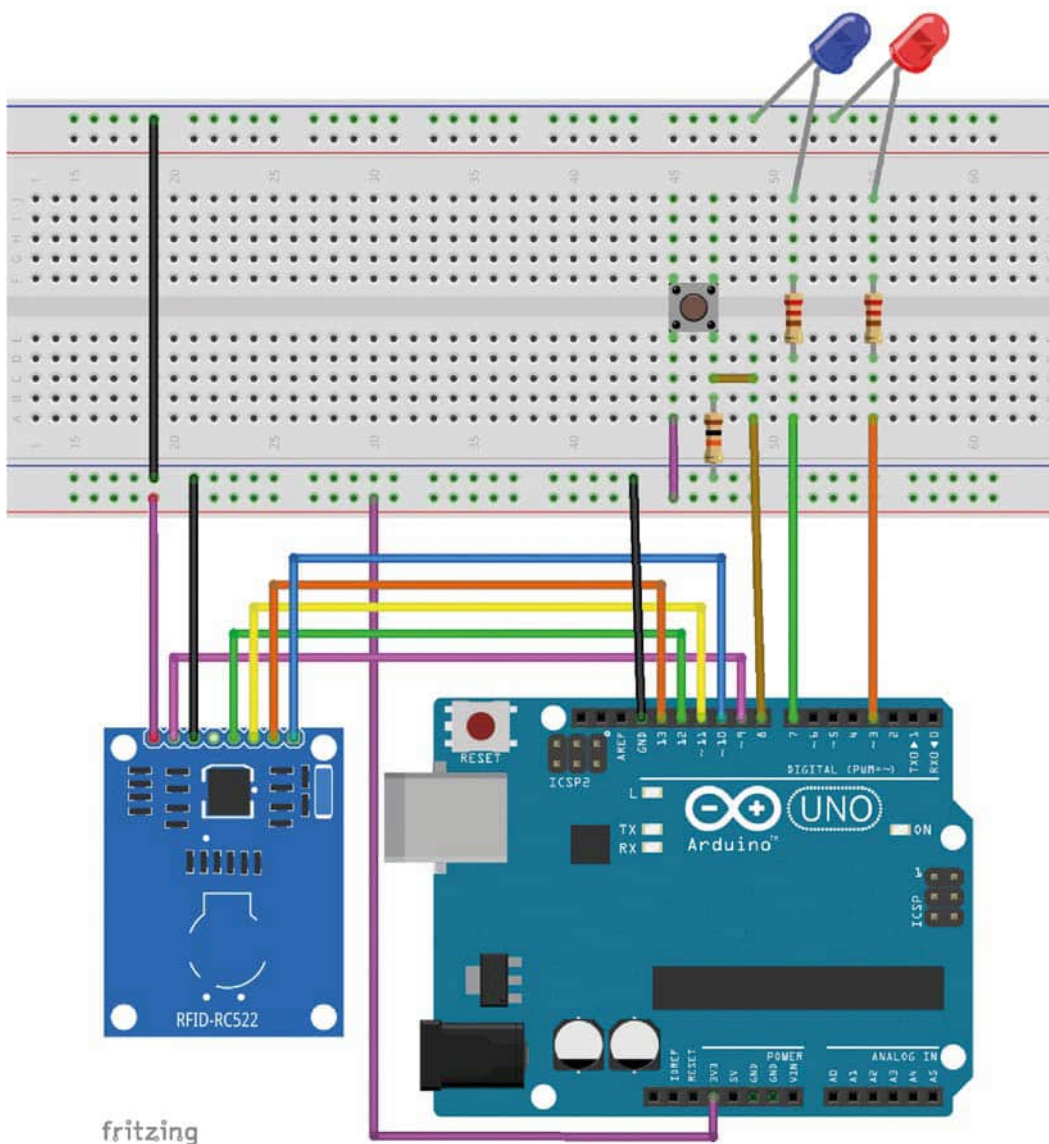
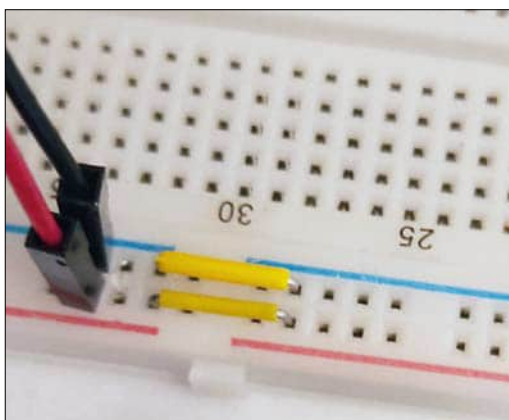


Fig. 7
Schema di cablaggio del controllo degli accessi.

→ **Fig. 8**
Particolare
del collegamento
delle piste
di alimentazione
della breadboard.



Il materiale occorrente a realizzare il progetto è:

- Modulo RFID-RC522;
- Arduino UNO;
- Breadboard con le linee di alimentazione laterali (rossa e nera oppure rossa e blu);
- Tastino 6x6mm;
- LED rosso;
- LED blu (o altro colore a scelta);
- n. 2 resistori da 220 Ohm;
- n. 1 resistori da 10 kOhm;
- n. 5 cavetti jumper M-M;
- n. 7 cavetti jumper M-F;
- n. 1 ponticello jumper passo 3;
- n. 1 ponticello jumper passo 5;
- n. 1 ponticello jumper passo 19;
- n. 4 Tag, di cui uno contrassegnato come amministratore.

Notate che riguardo a cavetti e ponticelli jumper, abbiamo fornito l'elenco per replicare esattamente il nostro schema pratico; naturalmente è possibile apportare qualsiasi variazione, a patto di eseguire correttamente i collegamenti proposti, in caso contrario il circuito non funzionerà; soprattutto bisogna fare estrema attenzione a non invertire

l'alimentazione (3,3V e GND) altrimenti si rischia di bruciare il modulo!

Una nota particolare riguarda la breadboard; nelle nostre prove ci è capitato un modello con le linee di alimentazione interrotte nella parte centrale, in realtà questo era chiaramente indicato dall'interruzione delle linee di serigrafia rosse e blu ma, essendo una cosa inusuale, inizialmente non ci abbiamo fatto caso, col risultato che il LED del modulo RC522 si accendeva lo stesso (a causa della corrente portata dai segnali) ma la lettura dei Tag non avveniva; abbiamo quindi dovuto aggiungere due ponticelli per collegare le due piste per l'intera lunghezza (**Fig. 8**) ossia per ponticellarle.

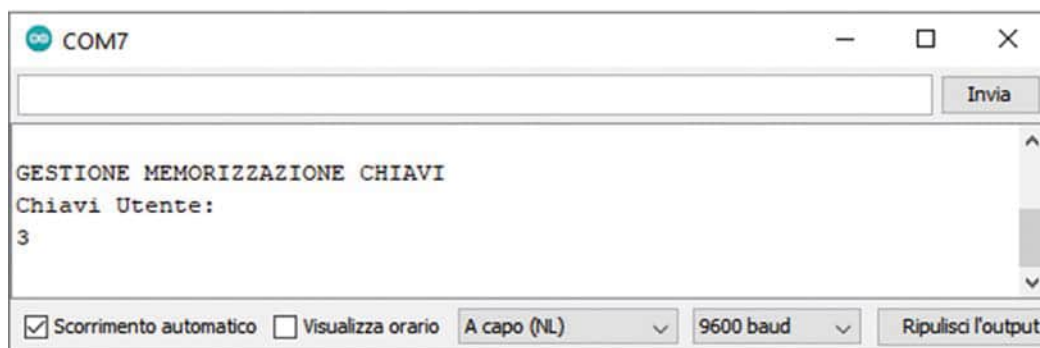
FUNZIONAMENTO DEL CONTROLLO DI ACCESSI

Una volta completato il circuito su breadboard e controllati i collegamenti, per iniziare sarà sufficiente ricollegare il cavetto USB proveniente dal PC alla board Arduino UNO e si vedrà l'accensione del LED rosso del modulo RC522 (ma come abbiamo spiegato prima, questo non è indicativo di corretta alimentazione) mentre i due LED rosso e blu sulla breadboard resteranno spenti. Ora bisogna aprire il Monitor Seriale dell'IDE di Arduino, ed apparirà il messaggio "Controllo accessi attivo"; la prima operazione da fare deve essere quella di avvicinare al modulo il Tag Amministratore, se è tutto a posto i due LED lampeggeranno contemporaneamente per tre volte e rimarranno accesi, inoltre il Monitor Seriale mostrerà il messaggio visibile in **Fig. 9**, ad indicare che è attiva la modalità Amministratore.

Se alla prima accensione appare un numero diverso da 0 (in genere sarà 255) occorre resettare l'EEPROM, sarà sufficiente premere il tastino montato sulla breadboard per ottenere il risultato desiderato (**Fig. 10**).

Dopo il reset si uscirà dalla modalità Amministratore e si spegneranno i LED; bisogna ripassare il

→ **Fig. 9**
Modalità
Amministratore.



Tag Amministratore per rientrare, poi sarà sufficiente avvicinare il Tag da autorizzare per memorizzarne l'UID (**Fig. 11**).

L'operazione si potrà effettuare per un massimo di tre Tag, poi non sarà possibile memorizzarne altri, a meno di non modificare adeguatamente lo sketch.

A questo punto possiamo vedere il comportamento del circuito in base al tipo di Tag che avviciniamo al lettore RFID; le possibilità sono due.

1. Tag non memorizzato: si accende il LED rosso per circa tre secondi, poi si spegne; l'autorizzazione viene negata.
2. Tag memorizzato: se il LED blu è spento si accende, segno che l'autorizzazione è stata concessa; il LED resta acceso finché non si passa lo stesso Tag o un altro Tag memorizzato, a quel punto si spegne.

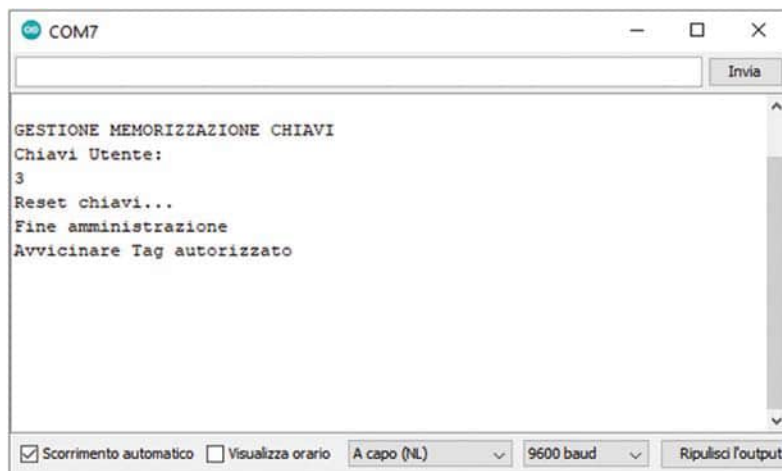
Un'ultima curiosità: in mancanza di altri Tag per simulare la non autorizzazione è possibile usare il proprio smartphone; se disponete di un cellulare dotato di chip NFC con Android 5.0 o versioni successive, sarà sufficiente attivare l'NFC nelle impostazioni e quindi usare lo smartphone come se fosse un Tag; infatti avvicinandolo al modulo dal lato del chip, esso sarà riconosciuto come Tag, ma con una particolarità che non ne permette l'uso come Tag autorizzato, infatti ad ogni nuova lettura l'UID risulta diverso rispetto alla lettura precedente; questa cosa è facilmente verificabile con lo sketch "RFID_TAG_CODE". Quindi se anche memorizzerete il chip come autorizzato, nelle letture successive non sarà comunque riconosciuto. In alcuni rari casi lo smartphone potrebbe aver bisogno di un'APP specifica per funzionare come Tag, sarà sufficiente scaricare dal Play Store "NFC Tools" e lasciarla attiva, senza fare altre operazioni.

CONCLUSIONI

Terminiamo questo articolo con un suggerimento: com'è facilmente immaginabile questo progetto si può ampliare aggiungendo, per esempio, un relé, pilotato tramite lo stesso pin del LED blu, che permetta di attivare un attuatore quale potrebbe essere una serratura elettrica, i limiti sono solo quelli della fantasia.

Naturalmente, maggiore sarà la complessità dell'ampliamento, maggiori saranno le variazioni ed implementazioni da fare allo sketch, ma sarà tutto un utile esercizio di programmazione che accrescerà le vostre competenze e che vi darà grande soddisfazione. 

 **Fig. 10**
Reset dell'EEPROM.



 **Fig. 11**
Memorizzazione di un Tag autorizzato.



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. Il lettore RFID con MFRC522 (cod. VMA405N) è in vendita a Euro 7,80. Le board Arduino sono disponibili sul sito www.futurashop.it. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Taglia e incidi con **Cricut MAKER**



Prezzo IVA inclusa

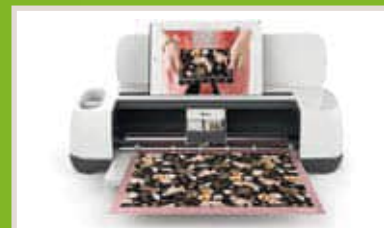
Macchina da taglio intelligente



€ 439,00

cod. CRICUTMAKER

Dai vita alla fantasia e alla creatività con Cricut Maker l'innovativa macchina da taglio. Grazie a Design Space, il software di progettazione gratuito, potrai realizzare praticamente qualsiasi progetto fai-da-te, dall'arte 3D all'arredamento per la casa, gioielli, progetti di carta e molto altro ancora e inviarti direttamente a Cricut Maker dal tuo tablet, dal PC o dallo smartphone. Cricut Maker taglia in modo rapido e accurato oltre 300 materiali: tessuti, pelle, carta, cartamodelli, cuoio, fogli di balsa e tanto altro. Espandi le potenzialità di Cricut Maker utilizzando utensili dedicati, alcuni inclusi nella confezione, altri acquistabili separatamente.





dell' Ing. GIOSUÈ CALIANO

LA CONVERSIONE A/D VELOCE PER FISHINO32

Campioniamo con un Fishino a 32 bit segnali analogici e calcoliamone lo spettro. Un OLED di piccole dimensioni ci mostrerà in tempo reale il risultato.

M

olti dei problemi di elaborazione del segnale possono essere risolti in modo efficiente mediante l'uso di tecniche digitali. Tuttavia, poiché il segnale spesso ha un'origine analogica (il mondo "reale" è analogico!) esso deve essere trasformato in una forma digitale per poter essere gestito dalle schede a microcontrollore con cui abbiamo dimestichezza. Per realizzare questa trasformazione dal mondo analogico a quello digitale sono state impiegate negli anni molte tecniche. In questo articolo esamineremo una classe di dispositivi, denominati convertitori analogico-digitali (ADC), molto comuni nell'elettronica moderna, e in particolare la nostra attenzione sarà focalizzata sui convertitori A/D che sono disponibili nelle schede tipo Arduino, e in particolare

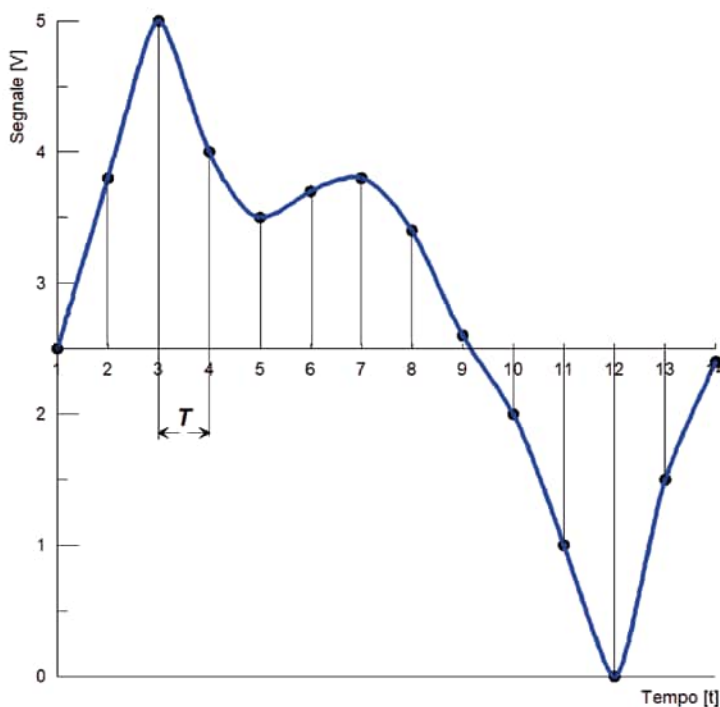


Fig. 1
Un segnale analogico con componente DC a 2,5 V: il tempo di campionamento è T e la frequenza di campionamento è $FS = 1/T$.

nelle più performanti schede della serie Fishino a 32 bit. Perché questa scelta che potrebbe sembrare riduttiva? Il motivo è relativo alla velocità di acquisizione massima che possiamo raggiungere e di conseguenza la massima frequenza dei segnali che possiamo acquisire e di cui possiamo visualizzare lo spettro.

L'Atmel ATmega328P, il cuore dell'Arduino Uno, è dotato, come tutti i core AVR, di un convertitore analogico-digitale a 10 bit. Un segnale analogico è un flusso continuo di valori analogici e può essere campionato un certo numero di volte al secondo, che normalmente viene indicata come frequenza di campionamento (FS).

La **Fig. 1** mostra un segnale analogico con una componente DC di circa 2,5 Volt, campionato nel tempo, con periodo di campionamento T che rappresenta la frequenza di campionamento ($FS = 1/T$). Esistono diversi tipi di architetture ADC, ma la maggior parte degli AVR a 8 bit utilizza l'ADC ad approssimazione successiva (SAR, Successive-Approximation Register). I convertitori ad approssimazione successiva sono ampiamente utilizzati, soprattutto nei microcontrollori, perché sono capaci di alta risoluzione e alta velocità. Il tempo di conversione è fisso e indipendente dall'entità della tensione di ingresso. Ogni conversione è unica e indipendente dai risul-

tati della conversione precedente, poiché la logica interna viene cancellata all'inizio di ogni conversione. La tecnica di conversione consiste nel confrontare l'ingresso sconosciuto con una tensione precisa generata da un convertitore digitale-analogico (DAC) interno al dispositivo. L'ingresso del DAC è il valore digitale dell'uscita dell'ADC.

Il processo di conversione è simile al processo di pesatura che si utilizza con una bilancia, con una serie di n pesi binari. Dopo che è stato inviato il comando di inizio della conversione e il valore della precedente conversione è stato cancellato, l'uscita del bit più significativo (MSB) del DAC è confrontata con l'ingresso. Se l'ingresso è maggiore del MSB, rimane acceso (cioè 1 logico) e viene provato il bit successivo. Se l'ingresso è inferiore a MSB, viene disattivato (ovvero viene impostato a 0 logico) e viene provato il bit successivo. Se il secondo bit non raggiunge il "peso" sufficiente per superare l'ingresso, viene lasciato sul valore "alto" (1) e viene provato il terzo bit. Il processo continua in ordine decrescente di bit fino a quando non è stato provato l'ultimo bit. Dopo che il processo è completato, il registro di stato indica che il contenuto del registro di uscita ora costituisce una conversione valida.

Il contenuto del registro di uscita forma un codice digitale binario corrispondente all'ampiezza del segnale di ingresso. Se l'ingresso dovesse cambiare durante la conversione, il dato di uscita non potrebbe più rappresentare l'ingresso analogico. Per evitare problemi di questo tipo, di solito si utilizza un dispositivo di "Sample & Hold" (SHA) prima della logica di conversione, per campionare il segnale e mantenere il valore dell'ingresso che era presente in un dato momento prima dell'inizio della conversione.

Nell'Atmel è presente un SHA che funziona anche da comparatore: esso mantiene un'uscita costante durante tutto il processo di conversione. L'uscita di stato del convertitore può essere utilizzata per rilasciare il SHA dalla sua modalità di attesa, al termine della conversione.

In **Fig. 2** riportiamo un diagramma semplificato dell'ADC AVR a 8 bit.

L'AVR ha solo un ADC a 10 bit e utilizza un multiplexer analogico a 8 canali per campionare ciascuno degli 8 canali analogici di ingresso. Il circuito ADC preleva i campioni e il SAR (Conversion Logic nella **Fig. 2**) ha un clock dedicato che garantisce l'indipendenza della conversione dagli altri compiti del microcontrollore. Il meccanismo di conversione può essere attivato su richiesta o automaticamente.

Il risultato della conversione a 10 bit viene memorizzato in due registri (ADCH e ADCL). Arduino utilizza un semplice comando per avviare una singola conversione: *analogRead (numero pin)*, dove "numero pin" indica il canale analogico di ingresso (A0-A7). Se non viene eseguito nulla altro, la conversione è completa. È necessario utilizzare frequenze di clock comprese tra 50 kHz e 200 kHz per ottenere la risoluzione più elevata. Supponiamo di avere un clock di sistema con frequenza di 16 MHz e di impostare il fattore di divisione del pre-scaler dedicato all'ADC (registro ADCSRA, fig. 2) su 128, che è l'impostazione predefinita di Arduino: la frequenza di clock dell'ADC risulta essere allora di circa 125 KHz. La conversione totale richiede 14,5 cicli di clock, per cui il tempo di conversione è pari a 116 μ s che corrisponde ad una frequenza di campionamento di 8,6 KHz, non eccessivamente elevata. Anche se impostiamo il pre-scaler ADCSRA al massimo valore possibile (cioè 2) al fine di ottenere il massimo valore della

frequenza di clock dell'ADC, si avrà un tempo di campionamento che non va al di sotto di 5,6 μ s che corrisponde ad una frequenza di campionamento massima di circa 178 KHz. Purtroppo questo limite non è raggiungibile. La frequenza di clock ADC massima consigliata è limitata dal DAC interno utilizzato per effettuare la conversione. Per le prestazioni ottimali, il clock dell'ADC non deve superare i 200 kHz (infatti il datasheet segnala una frequenza di campionamento massima di 15 KHz), tuttavia frequenze fino a 1 MHz non riducono la risoluzione dell'ADC, per cui il valore massimo consigliabile per il pre-scaler è 16 e quindi il tempo di conversione si riduce a circa 17 μ s che corrisponde ad una frequenza di campionamento massima di 58 KHz. Questo è il massimo valore che si può ottenere con un Arduino versione base.

CONVERSIONE A/D CON UN PIC32MX470

Come è intuibile, la conversione A/D effettuata

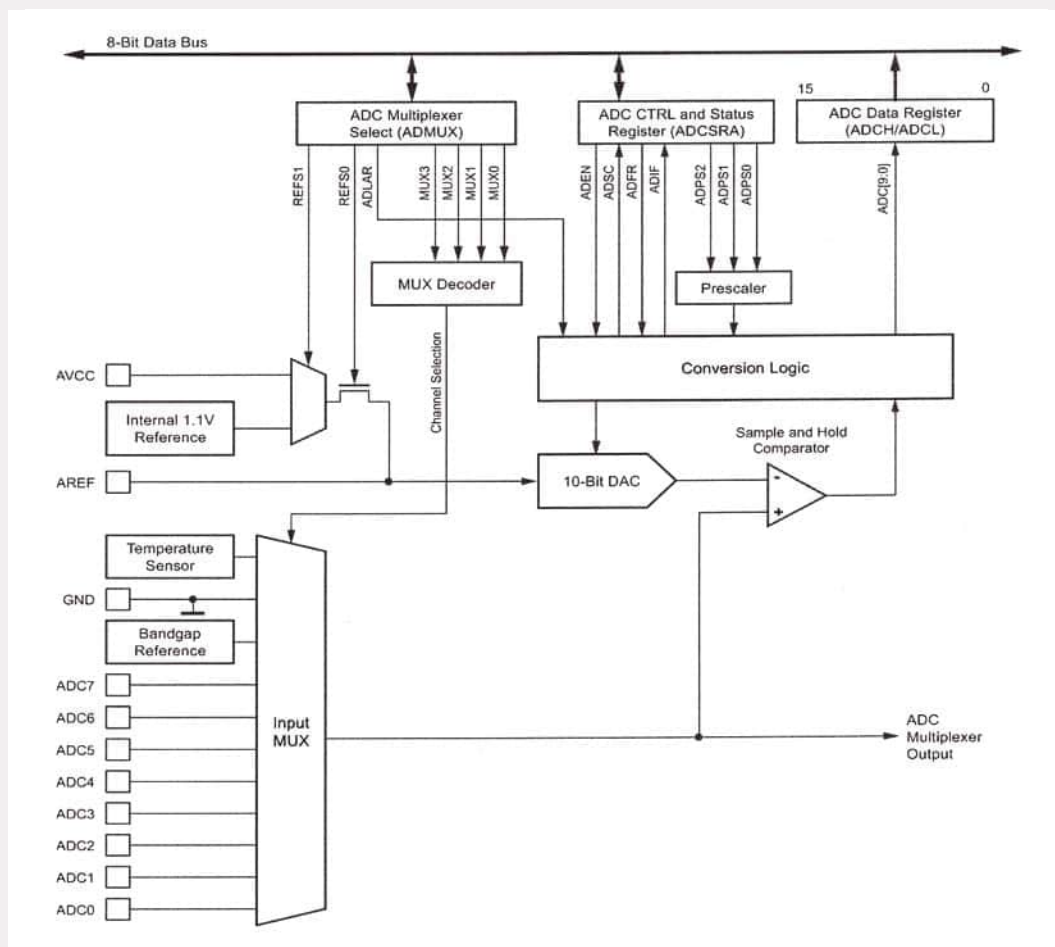
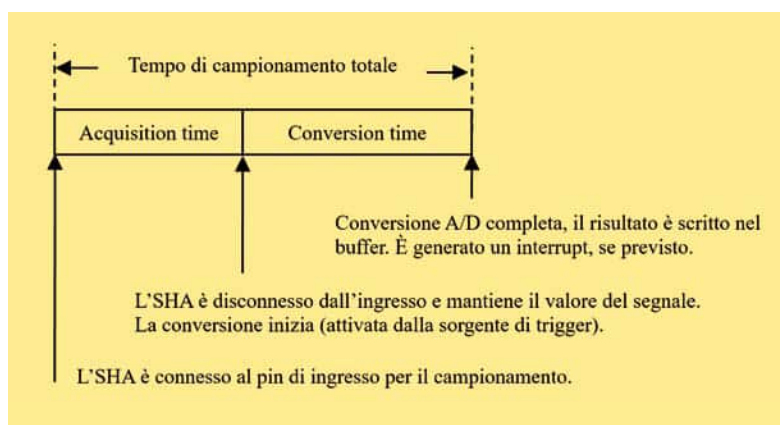


Fig. 2
Schema a blocchi dell'ADC dell'ATmega328P.



Fig. 3
Processo di
acquisizione e
conversione.



mediante un PIC32MX è sicuramente migliore dal punto di vista dei parametri in gioco. Infatti tale microcontrollore è certamente più performante rispetto all'ATmega328P, e quindi ci aspettiamo di avere delle caratteristiche di funzionamento della conversione di un livello superiore.

Infatti, già leggendo le caratteristiche peculiari dal datasheet osserviamo (tra le altre cose) che la velocità di conversione può arrivare fino a 1 Msps (1 mega sample al secondo)!

Riassumendo: 10 bit di conversione, 28 canali multiplexati, la possibilità di avere una tensione di riferimento esterna, un amplificatore "Sample and Hold" (S&H) sia in modalità unipolare che bipolare, la possibilità di effettuare una scansione automatica dei canali da convertire, la possibilità di scegliere una sorgente di trigger per la partenza

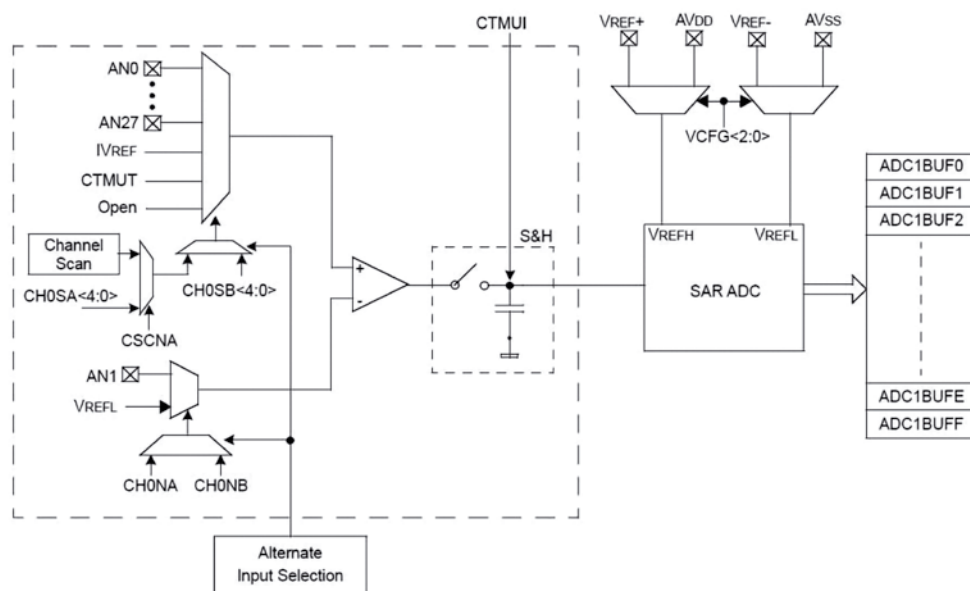
della conversione, un buffer che contiene il risultato della conversione composto da 16 locazioni a 16 bit, otto diversi formati di rappresentazione dei dati nel buffer dei risultati, la possibilità di operare anche quando la CPU è nelle modalità Sleep e Idle. Queste sono caratteristiche di assoluto rispetto, e considerando che la PIC32MX470 è a bordo delle schede Fishino32/Shark/Piranha, le prospettive per l'utente di queste schede sono assolutamente rosee rispetto ad altre schede equivalenti.

PANORAMICA DEL FUNZIONAMENTO

Il campionamento analogico con il PIC32 consiste in due fasi: la fase di acquisizione e quella della conversione (**Fig. 3**). Durante l'acquisizione, il pin di ingresso analogico è collegato all' SHA.

Dopo che il pin è stato campionato per un periodo

Fig. 4
Diagramma a
blocchi dell'ADC
del PIC32MX470.



AD1CHS – ADC Input Select Register

30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
U	U									U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	
BIT 31		CH0NB: Negative Input Select bit for SAMPLE B 1 = Channel 0 negative input is AN1 0 = Channel 0 negative input is VR-																													
BIT 28-24		CH0SB: Positive Input Select bits for Sample B 11110 = Channel 0 positive input is Open 11101 = Channel 0 positive input is CTMU temperature sensor (CTMUT) 11100 = Channel 0 positive input is IVREF 11011 = Channel 0 positive input is AN27 • • • 00001 = Channel 0 positive input is AN1 00000 = Channel 0 positive input is AN0																													
BIT 23		CH0NA: Negative Input Select bit for SAMPLE A Multiplexer Setting 1 = Channel 0 negative input is AN1 0 = Channel 0 negative input is VR-																													
BIT 20-16		CH0SA: Positive Input Select bits for Sample A Multiplexer Setting 11110 = Channel 0 positive input is Open 11101 = Channel 0 positive input is CTMU temperature sensor (CTMUT) 11100 = Channel 0 positive input is IVREF 11011 = Channel 0 positive input is AN27 • • • 00001 = Channel 0 positive input is AN1 00000 = Channel 0 positive input is AN0																													

*U = indefinito

ESEMPIO: AD1CHS = 0x000B0000;

Per SAMPLE B, il negativo (di riferimento) è scelto VR-

Per SAMPLE B, il positivo è l'ingresso AN0

Per SAMPLE A, il negativo (di riferimento) è scelto VR-

Per SAMPLE A, il positivo è l'ingresso AN11

Tabella 1
Registro di
selezione
degli ingressi
(AD1CHS).

sufficiente e la tensione di campionamento è equivalente all'ingresso, il pin di ingresso viene scollegato dall'SHA per fornire una tensione di ingresso stabile al processo di conversione. Il processo di conversione converte quindi la tensione del campione analogico in una rappresentazione binaria. Anche in questo caso il processo di conversione è delegato a un registro ad approssimazioni successive dello stesso tipo di quello visto prima per l'ATmega328P.

Facendo riferimento alla **Fig. 4**, osserviamo che l'ADC a 10 bit ha un singolo SHA. L'SHA è collegato ai pin di ingresso analogico tramite il multiplexer di ingresso analogico. Il multiplexer di ingresso analogico è controllato dal registro AD1CHS (vedi **Tabella 1**).

Ci sono due serie di bit di controllo (CH0SA e CH0SB) nel registro AD1CHS. Questi due set di bit di controllo consentono il controllo indipendente dei due diversi ingressi analogici, per poter utilizzare l'SHA in modalità unipolare/bipolare. L'ADC può anche eseguire la scansione facoltativa attraverso una serie di ingressi analogici utilizzan-

do un singolo multiplexer.

Il tempo di acquisizione (in **Fig. 3** è chiamato "Acquisition time") può essere controllato manualmente o automaticamente. Il tempo di acquisizione può essere avviato manualmente impostando il bit SAMP (nel registro AD1CON1) e terminato manualmente cancellando lo stesso bit, mediante il software utente.

Il tempo di acquisizione può essere anche avviato automaticamente dall'hardware ADC e terminato automaticamente da una sorgente di attivazione (trigger) della conversione. Il tempo di acquisizione è impostato dai bit SAMC nel registro AD1CON3 (bit 12-8). L'ADC richiede un ciclo di clock (TAD) per convertire ogni bit del risultato, più due cicli di clock aggiuntivi. Pertanto, sono necessari un totale di 12 cicli di clock TAD per eseguire la conversione completa. Quando il tempo di conversione è completo, il risultato viene scritto in uno dei 16 registri dei risultati ADC (da ADC1BUF0 a ADC1BUFF).

La somma del tempo di acquisizione e del tempo di conversione analogico-digitale fornisce il tempo di campionamento totale (**Fig. 3**).



Tabella 2
AD1CON1
Registro di controllo 1.

AD1CON1 – registro di controllo 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

BIT 15	ON: ADC Operating Mode bit 1 = ADC module is operating 0 = ADC is off
BIT 13	SIDL: Stop in Idle Mode bit 1 = Discontinue module operation when device enters Idle mode 0 = Continue module operation in Idle mode
BIT 10-8	FORM: Data Output Format bits 011 = Signed Fractional 16-bit 010 = Fractional 16-bit 001 = Signed Integer 16-bit 000 = Integer 16-bit 111 = Signed Fractional 32-bit 110 = Fractional 32-bit 101 = Signed Integer 32-bit 100 = Integer 32-bit
BIT 7-5	SSRC: Conversion Trigger Source Select bits 111 = Internal counter ends sampling and starts conversion (auto convert) 110 = Reserved 101 = Reserved 100 = Reserved 011 = CTMU ends sampling and starts conversion 010 = Timer3 period match ends sampling and starts conversion 001 = Active transition on INT0 pin ends sampling and starts conversion 000 = Clearing SAMP bit ends sampling and starts conversion
BIT 4	CLRASAM: Stop Conversion Sequence bit (when the first ADC interrupt is generated) 1 = Stop conversions when the first ADC interrupt is generated. Hardware clears the ASAM bit when the ADC interrupt is generated. 0 = Normal operation, buffer contents will be overwritten by the next conversion sequence
BIT 2	ASAM: ADC Sample Auto-Start bit 1 = Sampling begins immediately after last conversion completes; SAMP bit is automatically set 0 = Sampling begins when SAMP bit is set
BIT 1	SAMP: ADC Sample Enable bit 1 = The ADC SHA is sampling 0 = The ADC sample/hold amplifier is holding When ASAM = 0, writing '1' to this bit starts sampling. When SSRC = 000, writing '0' to this bit will end sampling and start conversion
BIT 0	DONE: Analog-to-Digital Conversion Status bit 1 = Analog-to-digital conversion is done 0 = Analog-to-digital conversion is not done or has not started Clearing this bit will not affect any operation in progress

*U = indefinito

ESEMPIO: AD1CON1 = 0x0000;

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC off (bit 15)

Operazione continua anche in Idle mode (bit 13)

Risultato della conversione come Integer 16-bit (bit 10-8)

Reset il SAMP bit alla fine della conversione e inizia la conversione (bit 7-5)

Operazione normale, il contenuto del buffer sarà sovrascritto alla prossima conversione (bit 4)

Il campionamento inizia quando il bit SAMP è posto a 1 (bit 2).

L'amplificatore SHA dell'ADC è "holding" (bit 1)

La conversione analogico/digitale non è terminata oppure non è iniziata (bit 0).

Sono disponibili più opzioni per il clock di input dell'ADC, utilizzate per creare "l'orologio TAD". L'utente deve selezionare un'opzione di clock di input che non violi la specifica della TAD minima. Il processo di campionamento può essere eseguito una volta, periodicamente o in base a un trigger come definito dalla configurazione del modulo.

L'inizio del campionamento può essere controllato nel software impostando il bit SAMP (AD1CON1). L'inizio del tempo di campionamento può anche essere controllato automaticamente dall'hardware. Quando l'ADC funziona in modalità di campionamento automatico, l'SHA viene ricollegato al pin di ingresso analogico alla fine della conversione nella



AD1CON2 – registro di controllo 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

BIT 15-13	VCFG: Voltage Reference Configuration bits		
		ADC V_{R+}	ADC V_{R-}
	000	AVDD	AVSS
	001	External AVref+ pin	AVSS
	010	AVDD	External AVref- pin
	011	External AVref+ pin	External AVref- pin
	1xx	AVDD	AVSS
BIT 12	OFFCAL: Input Offset Calibration Mode Select bit 1 = Enable Offset Calibration mode Positive and negative inputs of the sample and hold amplifier are connected to VREFL 0 = Disable Offset Calibration mode The inputs to the sample and hold amplifier are controlled by AD1CHS or AD1CSSL		
BIT 10	CSCNA: Input Scan Select bit 1 = Scan inputs 0 = Do not scan inputs		
BIT 7	BUFS: Buffer Fill Status bit. Only valid when BUFM = 1 (ADRES split into 2 x 8-word buffers). 1 = ADC is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7 0 = ADC is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF		
BIT 5-2	SMPI: Sample/Convert Sequences Per Interrupt Selection bits 1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence 1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence • • • 0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence 0000 = Interrupts at the completion of conversion for each sample/convert sequence		
BIT 1	BUFM: ADC Result Buffer Mode Select bit 1 = Buffer configured as two 8-word buffers, ADC1BUF(7...0), ADC1BUF(15...8) 0 = Buffer configured as one 16-word buffer ADC1BUF(15...0).		
BIT 0	ALTS: Alternate Input Sample Mode Select bit 1 = Uses Sample A input multiplexer settings for first sample, then alternates between Sample B and Sample A input multiplexer settings for all subsequent samples 0 = Always use Sample A input multiplexer settings		

*U = indefinito

ESEMPIO: AD1CON2 = 0x0000;

Viene usata la tensione AVDD e AVSS per VR+ e VR-, rispettivamente.

Disabilitato il modo di calibrazione dell'offset.

Non viene effettuata la scansione degli ingressi.

Il convertitore utilizza il buffer 0x0-0x7 per memorizzare il dato, mentre l'utente accede ai dati nei registri 0x8-0xF.

L'interrupt avviene al completamento della sequenza di conversione.

AD1CON3 – registro di controllo 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

BIT 15	ADRC: ADC Conversion Clock Source bit 1 = Clock derived from FRC 0 = Clock derived from Peripheral Bus Clock (PBCLK)
BIT 12-8	SAMC: Auto-sample Time bits 11111 = 31 TAD - - - 00010 = 2 TAD 00001 = 1 TAD 00000 = 0 TAD (Not allowed)
BIT 7-0	ADCS: ADC Conversion Clock Select bits 11111111 TAD = TPB • 2 • (ADCS + 1) = 512 • TPB - - - 00000011 TAD = TPB • 2 • (ADCS + 1) = 8 • TPB 00000010 TAD = TPB • 2 • (ADCS + 1) = 6 • TPB 00000001 TAD = TPB • 2 • (ADCS + 1) = 4 • TPB 00000000 TAD = TPB • 2 • (ADCS + 1) = 2 • TPB

*U = indefinito

ESEMPIO: AD1CON3 = 0x020F;

Il clock di conversione è ricavato dal clock del bus delle periferiche (PBCLK)

L'auto-sample time bit è settato a 2*TAD

Tabella 3
AD1CON2
Registro di controllo 2.

Tabella 4
AD1CON3
Registro di controllo 3.

Tabella 5
Registro di
selezione della
scansione
automatica dei
canali (AD1CSSL)

AD1CSSL: ADC INPUT SCAN SELECT REGISTER

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U																															

BIT 30-0	CSSL: ADC Input Pin Scan Selection bits 1 = Select ANx for input scan 0 = Skip ANx for input scan CSSL = ANx, where x = 0-27; CSSL30 selects Vss for scan; CSSL29 selects CTMU input for scan; CSSL28 selects IVREF for scan.
----------	---

*U = indefinito

ESEMPIO: **AD1CSSL = 0**

Non viene effettuato lo "scan" degli ingressi.

ESEMPIO: **AD1CSSL = 0x000F**

Viene effettuato lo "scan" degli ingressi, da AN0 ad AN3

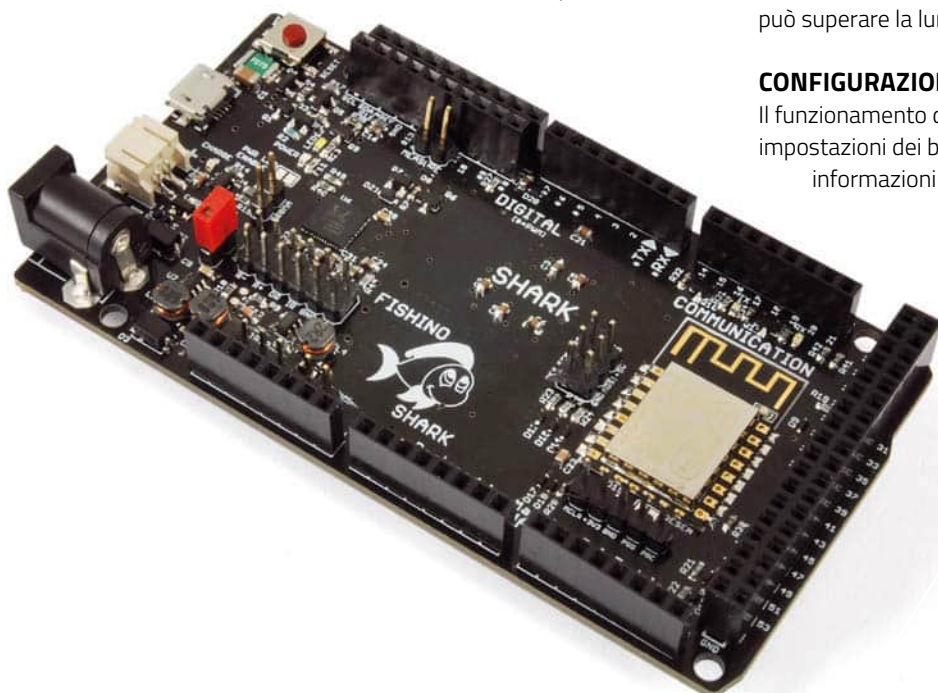
sequenza di campionamento/conversione. La funzione di campionamento automatico è controllata dal bit ASAM del registro AD1CON1. La sorgente del trigger di conversione termina il tempo di campionamento e inizia una conversione da analogico a digitale o una sequenza di campionamento/conversione. La sorgente del trigger di conversione viene selezionata dai bit SSRC del registro AD1CON1. Il trigger di conversione può essere scelto tra una varietà di sorgenti hardware o può essere controllato manualmente nel software cancellando il bit SAMP. Una delle sorgenti di attivazione della conversione è la conversione automatica. Il tempo tra le conversioni automatiche è impostato da un contatore e dal clock dell'ADC. La modalità di campionamento

automatico e il trigger di conversione automatica possono essere utilizzati insieme per fornire conversioni automatiche infinite senza l'intervento del software.

È possibile generare un interrupt alla fine di ciascuna sequenza di campioni o di più sequenze di campioni come determinato dal valore dei bit SMPI del registro AD1CON2. Il numero di sequenze di campioni tra gli interrupt può variare tra 1 e 16. L'utente deve tenere presente che il buffer di conversione analogico-digitale contiene i risultati di una singola sequenza di conversione. La sequenza successiva inizia a riempire il buffer dall'alto anche se il numero di campioni nella sequenza precedente era inferiore a 16. Il numero totale di risultati di conversione tra interrupt è il valore SMPI. Il numero totale di conversioni tra interrupt non può superare la lunghezza del buffer fisico.

CONFIGURAZIONE TIPICA DEL MODULO ADC

Il funzionamento del modulo ADC è regolato dalle impostazioni dei bit nei 5 registri di cui diamo informazioni dettagliate nelle relative tabelle.



Le seguenti istruzioni riassumono le azioni e le impostazioni più comuni.

Ovviamente le operazioni relative allo stesso registro vengono effettuate simultaneamente, per cui anche se sembra complesso, le operazioni si esauriscono scrivendo nei 5 registri dedicati all'ADC. Nel software che proponiamo come esempio queste operazioni sono raccolte nella procedura *init_adc()*. Per configurare il modulo ADC, occorre eseguire le seguenti operazioni:

1. Selezionare gli ingressi analogici al multiplexer ADC in AD1CHS.
2. Selezionare il formato del risultato dell'ADC utilizzando i bit FORM nel registro AD1CON1.
3. Selezionare la sorgente del clock utilizzando il campo SSRC nel registro AD1CON1.
4. Selezionare la sorgente del riferimento di tensione utilizzando il VCFG del registro AD1CON2.
5. Selezionare la modalità di scansione (o meno) utilizzando il bit CSCNA del registro AD1CON2.
6. Impostare il numero di conversioni per ogni interrupt da generare, campo SMPI di AD1CON2, se si prevede di utilizzare gli interrupt.
7. Impostare la modalità di riempimento del buffer utilizzando il bit BUFM di AD1CON2.

VALORI AD1CON3 PER FISHINO SHARK

AD1CON3 (2*TAD)	T _{ADC} [μS]	f _c [KHZ]
0x0202	1.2	833
0x0203	1.6	625
0x0204	2.0	500
0x0205	2.4	416
0x0206	2.8	357
0x0207	3.2	312
0x0208	3.6	277
0x0209	4.0	250
0x020A	4.4	227
0x020B	4.8	208
0x020C	5.2	192
0x020D	5.6	178
0x020E	6.0	166
0x020F	6.4	156
.....		
0x0219	10.4	96
0x0233	20.8	48
0x0237	22.6	44.1

CALCOLO DELLA FREQUENZA DI CAMPIONAMENTO

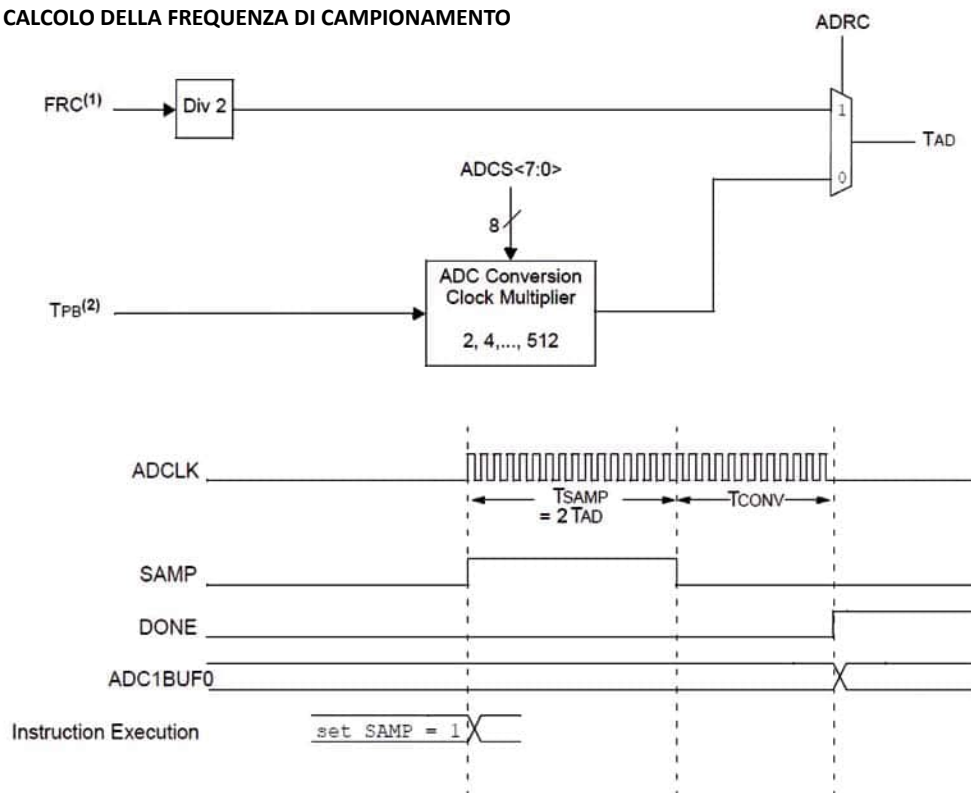
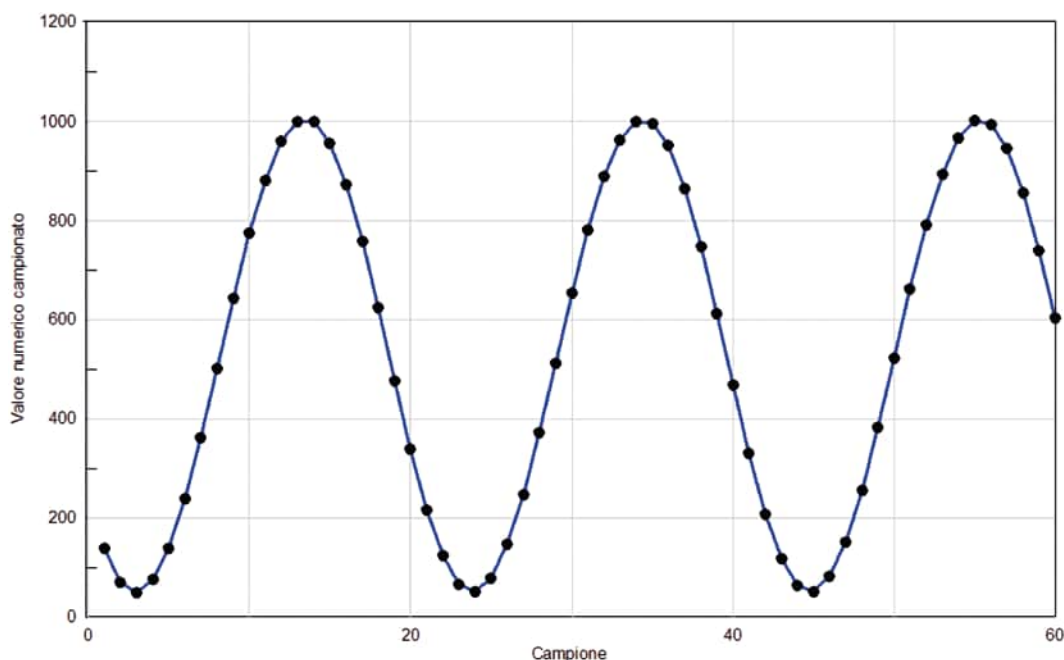


Fig. 5
Diagramma a blocchi relativo alla frequenza di campionamento dell'ADC del PIC32MX470 e relativo diagramma temporale.

Fig. 6
Esempio di campionamento a circa 500 KHz di una sinusoidale a 20 KHz mediante l'ADC del Fishino Shark, ottenuto utilizzando il Listato 1.



8. Selezionare il MUX da collegare all'ADC in ALTS di AD1CON2.
9. Selezionare la sorgente del clock ADC utilizzando ADRC nel registro AD1CON3.
10. Selezionare il tempo di campionamento utilizzando il campo SAMC nel registro AD1CON3, se si desidera utilizzare la conversione automatica.
11. Selezionare il pre-scaler del clock ADC utilizzando il campo ADCS del registro AD1CON3.
12. Accendere il modulo ADC utilizzando il bit 15 del registro AD1CON1.
13. Per configurare l'interrupt ADC (se necessario):
 - a) Azzerare il bit AD1IF (nel registro IFS1)
 - b) Selezionare la priorità di interrupt ADC mediante il campo AD1IP nel registro IPC e la priorità secondaria nel campo AD1IS sempre del registro IPC, qualora si utilizzano gli interrupt.
14. Avviare la sequenza di conversione avviando il campionamento

Nota: i passaggi da 1 a 11 sopra riportati possono essere eseguiti in qualsiasi ordine, ma il passaggio 12 deve essere il passaggio finale in ogni caso. Facciamo riferimento alla **Fig. 5**, e supponiamo che la frequenza principale della PIC sia FRC=20 MHz. Come si osserva dallo schema a blocchi della figura, esso è inizialmente diviso per 2 dalla porta logica. Siccome la frequenza del bus periferiche (FPB) è la stessa del clock principale (20 MHz) allora TPB = 50 ns. Il campo ADCS del registro AD1CON3 stabilisce quale sarà il moltiplicatore del TPB di

partenza per calcolare il tempo di conversione analogico/digitale (TAD).

Per esempio se scriviamo nel registro AD1CON3 il valore 0x20f, questo significa che abbiamo scelto per ADCS il valore TAD=32 TPB e per SAMC (il tempo di campionamento) il valore 2*TAD. Cioè possiamo stabilire sia il tempo del SHA (tempo di campionamento) che il tempo di conversione. Allora il tempo di conversione analogico digitale totale T_{ADC} si può calcolare semplicemente con

$$T_{ADC} = 2 \cdot [(ADCS + 1) \cdot 2] \cdot (SAMC) \cdot T_{PB}$$

E nel caso dell'esempio (AD1CON3 = 0x20f) otteniamo $T_{ADC} = 2 \cdot [(15+1) \cdot 2] \cdot (2) \cdot 50ns = 6.4 \mu s$. La tabella riportata nel box illustra i tempi di conversione impostabili (quando si sceglie per SAMC un valore di 2*TAD) e di conseguenza la frequenza di campionamento. In **Fig. 6** è riportata l'acquisizione effettuata con lo Shark per un segnale sinusoidale di circa 3,2 Vpp di ampiezza, a frequenza 20 KHz. È stata usata il valore di AD1CON3 = 0x204, cioè un tempo di campionamento totale di 2 μs (500 KHz). In effetti si osserva che invece di ottenere un numero di campioni pari a 25 (il periodo di una sinusoide a 20 KHz è di 50 μs), se ne ottengono solo 22, ciò vuol dire che la frequenza di campionamento è leggermente sovrastimata, in realtà abbiamo campionato a circa 440 KHz. Questo è dovuto essenzialmente a dei cicli macchina aggiuntivi che servono sia per memorizzare il dato

📌 Listato 1 - Firmware per acquisire un segnale e memorizzarlo nella SdCard.

```

/*****
This example shows how to use Fishino Shark as a datalogger on SdCard
Feel free to apply it to any other example. It's simple!
Author: G. Caliano
*****/
#include <Fishino.h>
#include <Wire.h>
#include <SPI.h>
#include "FishinoSdFat.h"
SdFat SD;
// SD chip select pin. Be sure to disable any other SPI devices.
#ifdef SDCS
#define SD_CS_PIN SDCS
#else
#define SD_CS_PIN SS
#endif
File myfile;

#define CHANNEL2 A1 // The signal to be acquired is connected to A1 pin.
int ledPin=13;
const uint16_t samples = 1024;
double V[samples];

void init_adc() {
  AD1CON1CLR = 0x8000; // disable ADC before configuration
  AD1CHS = 0x000B0000;
  /* 0x000B0000 Connect RB11/AN11 as CH0 input --> A1 per Fishino Shark; 0x000D0000 Connect RB13/AN13 as CH0
input --> A1 per Fishino32*/
  AD1CON1 = 0x0000; // SAMP bit = 0 ends sampling
  AD1CON2 = 0; // Select external VREF+ and VREF- pins
  AD1CON3 = 0x0204;
  /* Sample time 0x0204 2 us (500 kHz); 0x020f 6.4us; 0x021f 12.8us; 0x020b 4.8us (200KHz); 0x0232 10us
(100KHz)*/
  AD1CSSL = 0; // no scan
  AD1CON1SET = 0x8000; // turn on the ADC
}

void setup(void) {
  Serial.print("Initializing SD card...");
  if (!SD.begin(SD_CS_PIN, SD_SCK_MHZ(15))) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  init_adc();
}

void loop(void) {
  for(int i=0; i<samples; i++) V[i]=0;
  digitalWrite(ledPin, HIGH);

  noInterrupts();
  for(int i=0; i<samples; i++)
  {
    // V[i] = analogRead(CHANNEL2); Vecchio metodo
    AD1CON1SET = 0x0002; // start sampling ...
    AD1CON1CLR = 0x0002; // start Converting
    while (!(AD1CON1 & 0x0001)); // conversion done?
    V[i] = ADC1BUF0;
  }
  interrupts();
  // crea file
  String temp = "V_acq";
  temp.concat(".txt");
  char filename[temp.length()+1];
  temp.toCharArray(filename, sizeof(filename));
  myfile = SD.open(filename, FILE_WRITE);

  // registra su SD il segnale del sensore di forza
  if (myfile) {
    for (int aa = 0; aa < samples; aa++) {
      myfile.print(V[aa]);
      myfile.println();
    }
  } else
  {
    // if the file didn't open, print an error:
    Serial.println("error opening file");
  }
  delayMicroseconds(200);
  myfile.close();
  digitalWrite(ledPin, LOW);
  while (1) {}
}

```

📌 Listato 2 - Firmware per l'analizzatore di spettro

```
/******  
This example shows how to use Fishino Shark as a spectrum analyzer  
Feel free to apply it to any other example. It's simple!  
Author: G. Caliano  
*****  
#include <Fishino.h>  
#include <SPI.h>  
#include <U8g2lib.h>  
#include "fix_fft.h"  
  
// display set up, bar, line position L & R  
#define LINEY 50  
#define LINEXL 0  
#define LINEXR 128  
#define SAMPLES 128  
#define AUDIO A1  
  
U8G2_SSD1306_128X64_NONAME_1_HW_I2C u8g2(U8G2_R0);  
  
char im[SAMPLES];  
char data[SAMPLES];  
int barht[SAMPLES];  
  
void init_adc() {  
  AD1CON1CLR = 0x8000;    // disable ADC before configuration  
  AD1CHS = 0x000B0000;  
  /* 0x000B0000 Connect RB11/AN11 as CH0 input --> A1 per Fishino Shark; 0x000D0000 Connect RB13/AN13 as CH0  
  input --> A1 per Fishino32*/  
  AD1CON1 = 0x0000;      // SAMP bit = 0 ends sampling  
  AD1CON2 = 0;           // Select external VREF+ and VREF- pins  
  AD1CON3 = 0x0233;      // Sample time 0x0233 20.3us (48KHz)  
  AD1CSSL = 0;           // no scan  
  AD1CON1SET = 0x8000;   // turn on the ADC  
}  
  
void setup(void) {  
  u8g2.begin();  
  init_adc();  
}  
  
void loop(void) {  
  static int i, j;  
  int val;  
  // get audio data  
  noInterrupts();  
  for(i = 0; i < SAMPLES; i++)  
  {  
    AD1CON1SET = 0x0002;      // start sampling ...  
    AD1CON1CLR = 0x0002;     // start Converting  
    while (!(AD1CON1 & 0x0001)); // conversion done?  
    val = ADC1BUF0;  
    data[i] = (char)(val/4 - 128); // store as char  
    im[i] = 0;                // init all as 0  
  }  
  interrupts();  
  
  // run FFT  
  fix_fft(data, im, 7, 0);  
  // extract absolute value of data only, for 64 results  
  for(i = 0; i < SAMPLES/2; i++)  
  {  
    barht[i] = (int)sqrt(data[i] * data[i] + im[i] * im[i]);  
  }  
  for(i = 0, j = 0; i < SAMPLES/2; i++, j += 2)  
  {  
    barht[i] = barht[j] + barht[j + 1];  
  }  
  // display bargraph  
  bargraph(SAMPLES/4, barht); // plot SAMPLES / 4 = 32 as bargraph gen cannot handle 128 bars  
}  
// plot line and bar at position and height  
void bargraph(int n, int bh[])  
{
```



```

int i, s, w;                                     // bars, spacing and width
s = (LINEXR - LINEXL) / n;
w = s / 2;

u8g2.firstPage();
do
{
  u8g2.setFont(u8g_font_helvR08);
  u8g2.drawLine(LINEXL, LINEY, LINEXR, LINEY);
  u8g2.drawStr(0, LINEY + 10, "0");
  u8g2.drawStr(23, LINEY + 10, "5");
  u8g2.drawStr(47, LINEY + 10, "10");
  u8g2.drawStr(75, LINEY + 10, "15");
  u8g2.drawStr(105, LINEY + 10, "[kHz]");
  for(i = 0; i < n; i++)
  {
    u8g2.drawBox(LINEXL + s * i, LINEY - bh[i], w, bh[i] + 1); // u8glib doesn't accept box height
    of 0
  }
}
while(u8g2.nextPage());
}

```

nel buffer, sia per eseguire le altre istruzioni del ciclo di acquisizione, che vedremo nel prossimo paragrafo.

ESEMPIO DI ISTRUZIONI PER FISHINO

Vediamo come impostare le istruzioni nell'IDE di Arduino per realizzare quanto detto nei paragrafi precedenti. Dapprima scriviamo una procedura (*init_adc()*) per inizializzare i valori nei registri del convertitore analogico digitale della PIC32. Scegliamo nel registro AD1CHS il canale (o i canali) da utilizzare nel processo di conversione, stando attenti

alla tipologia di Fishino che utilizziamo (verificate lo schema elettrico della scheda da utilizzare, disponibile sul sito www.fishino.it), poiché alcuni pin analogici sono connessi in maniera diversa su Shark/32/Piranha.

Nel registro di controllo 1 (AD1CON1) scegliamo di operare anche in Idle mode oppure no, scegliamo il formato numerico della conversione all'interno del buffer (per esempio 16-bit integer), scegliamo di resettare il SAMP bit alla fine della conversione e impostiamo la modalità manuale di inizio

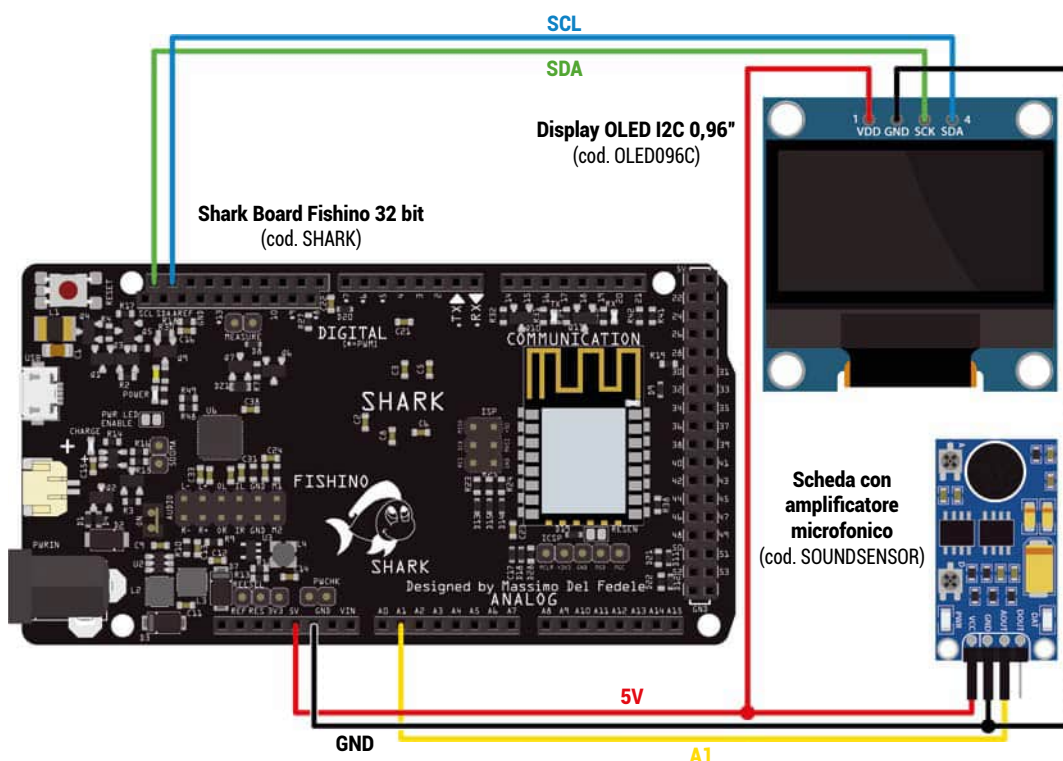


Fig. 7
Esempio di connessione per realizzare un analizzatore di spettro audio.

→ **Fig. 8**
Esempio di un
segnale a 10 kHz
(a sinistra) e di un
segnale a 5 kHz (a
destra) visualizzati
sull'Oled.



conversione sempre mediante il SAMP bit, infine scegliamo la modalità "holding" per il SHA. Poi passiamo ad impostare il registro di controllo 2 (AD1CON2), e in particolare scegliamo quali siano le tensioni di riferimento per la conversione, se calibriamo o meno l'offset, se facciamo una scansione degli ingressi oppure no, passiamo delle informazioni relative a come predisporre il buffer che conterrà i risultati della conversione e infine decidiamo relativamente all'interrupt che potrà essere generato alla fine della conversione. Nel registro di controllo 3 (AD1CON3) invece scegliamo i parametri relativi alla tempistica di acquisizione, di cui abbiamo già ampiamente parlato nel paragrafo precedente. Infine, nel registro AD1CSSL impostiamo l'eventuale scansione degli ingressi (oppure nessuna scansione). Alla fine accendiamo il convertitore settando il relativo bit del registro AD1CON1SET. In effetti è più lungo da descrivere che da fare, poiché si tratta di scrivere solo 6 registri! L'acquisizione invece la possiamo implementare ovviamente in qualsiasi parte del nostro software. Per ottimizzare i tempi di acquisizione possiamo utilizzare il costrutto all'interno di una struttura *noInterrupts()/Interrupts()* che ci permette di evitare possibili cause di ritardo per altre evenienze che possono interrompere casualmente il processo di conversione. All'interno di un ciclo (se necessario) occorre dapprima far partire il campionamento mediante la scrittura nel registro AD1CON1SET e quindi la conversione (sempre mediante AD1CON1CLR) e infine "ciclare" fino a che non è completata la conversione, testando il bit relativo di AD1CON1. Il valore numerico campionato lo ritroviamo nella prima posizione del buffer (ADC1BUF0) e lo possiamo assegnare alla variabile prevista. Tutto qui!

Nel **Listato 1** trovate una semplice procedura per acquisire un segnale dal pin A1 dello Shark, campionando 1024 valori dello stesso e memorizzandolo infine nella SdCard per eventuali successive elaborazioni. La **Fig. 6** è stata ottenuta mediante

questo firmware e il segnale è stato disegnato con un software su PC (Grapher della Golden Software). Infine, facendo riferimento alla **Fig. 7** e al **Listato 2**, mettiamo in pratica gli argomenti di cui abbiamo parlato. Realizziamo un piccolo ma interessante circuito che mostra lo spettro di un segnale analogico captato da una piccola scheda microfonica preamplificata, il cui software di base si trova facilmente in rete. Lo spettro lo mostriamo su un display Oled da 0.96" (vedi **Fig. 8**) che ci permette di utilizzare le caratteristiche grafiche di cui è dotato. Collegando i due moduli al Fishino Shark (ma si può benissimo usare il Fishino 32 oppure il Piranha) come mostrato in **Fig. 7**, possiamo visualizzare uno spettro a righe che copre certamente bene lo spettro del suono fino a 20 kHz (campioniamo a 48 kHz), ma possiamo anche spingerci oltre, considerando la massima frequenza di campionamento di cui abbiamo ampiamente parlato. Per generare un segnale di test si può usare una semplice app per cellulare (per esempio, per Android, si può usare "Simple Tone Generator"), e si possono generare segnali fino a circa 15 kHz.

Come al solito, queste informazioni e lo schema vogliono solo essere lo spunto per realizzare semmai progetti molto più complessi che utilizzano come base le informazioni qui riportate. □



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board Fishino 32 (cod. FISHINO32) è in vendita a Euro 65,00, la board Piranha (cod. PIRANHA) è disponibile a Euro 49,00, il display OLED (cod. OLED096C) costa Euro 11,00, il sensore con microfono (cod. SOUNDESENSOR) è disponibile a Euro 4,70. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

GENERATORE 100Hz

di **GUGLIELMO
BRAGUGLIA**

Generiamo una frequenza stabile e precisa utilizzando e confrontando la soluzione a logica cablata e quella basata su microcontrollore.

L

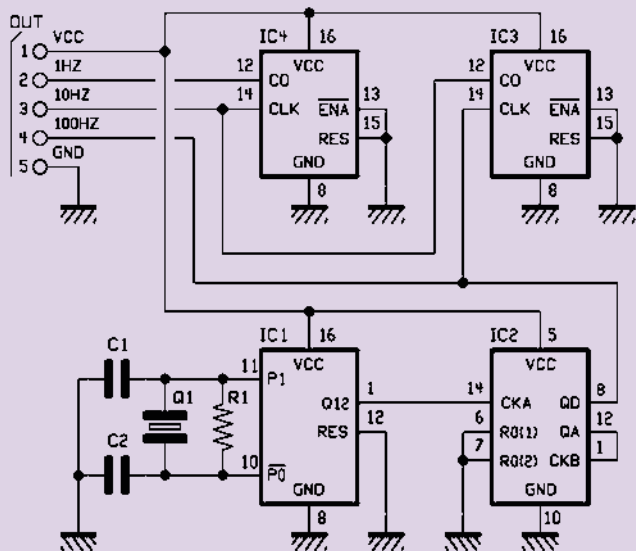
eggendo le richieste dei nostri lettori e visitando qualche forum abbiamo notato come una richiesta molto comune sia quella di avere un circuito in grado di generare una temporizzazione piuttosto precisa a 100Hz, tipicamente per effettuare misure del tempo con precisione al centesimo di secondo.

Per ottenere ciò esistono varie soluzioni, una delle quali è utilizzare un RTC per ricavare una frequenza precisa e stabile. La maggior parte degli RTC, oltre a svolgere il proprio compito, è in grado di fornire in uscita un segnale a 32.768Hz ma, anche utilizzando i sottomultipli che si ottengono dividendo mano mano per due, sino ad arrivare ad un segnale ad 1Hz, non corrispondono mai ai 100Hz precisi.

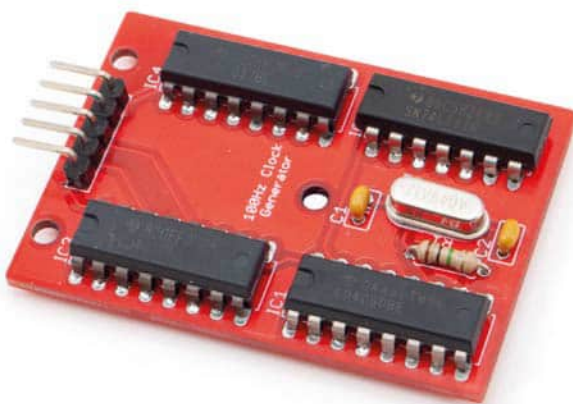
Una soluzione efficace e molto economica consiste nel non usare un RTC ma, piuttosto, un semplicissimo generatore a 100 Hz quarzato (e quindi con la dovuta precisione) da collegare al circuito che lo necessita: un generatore di onda rettangolare facilmente autocostruibile con pochi componenti elettronici, che qui vi proponiamo in una duplice veste, ossia in due versioni e altrettante filosofie progettuali.

schema elettrico

GENERATORE A LOGICA

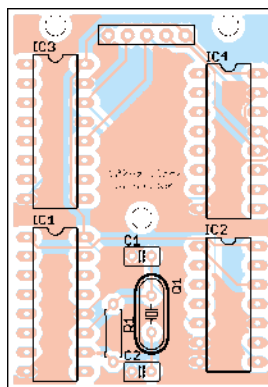
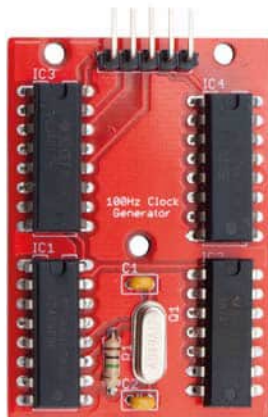


Il circuito del generatore a logica cablato (generatore 1.)



piano di montaggio

GENERATORE A LOGICA



Elenco Componenti:

R1: 1 Mohm
C1, C2: 20 pF
Q1: 4.9152 MHz
IC1: 4060N
IC2: 74LS92N
IC3, IC4: 4017N

Varie:

- Strip maschio 5 poli 90°
- Stampato S1570 (51x36 mm)

Un oscillatore del genere non serve solamente come riferimento generico per le misure al centesimo di secondo, ma potrebbe anche essere utilizzato, collegandolo ad un pin di interrupt di una delle vecchie schede Arduino (quelle non dotate di oscillatore quarzato, ma solo di un risuonatore ceramico) per ottenere dei tempi precisi al centesimo di secondo scrivendosi, ad esempio, una piccola gestione dell'interrupt ed associata funzione che, con molta fantasia, ricalcando il nome della **millis()**, si potrebbe chiamare **cents()**.

Per la realizzazione del generatore di 100Hz proporrò due soluzioni, una basata su componenti discreti (con i classici IC della serie 40xx e 74xx) con un quarzo a 4.9152 MHz, che chiameremo per praticità GENERATORE A LOGICA, ed una (che chiameremo GENERATORE CON MCU) basata su un piccolo ed economico microcontrollore della stessa famiglia AVR, l'Attiny24, con un quarzo a 8MHz; i due quarzi sono facilmente reperibili anche con ottima tolleranza (≤ 20 ppm).

GENERATORE A LOGICA

Vediamo per primo come realizzare un semplice generatore di segnale a 100Hz utilizzando un paio di comunissimi circuiti integrati dove, volendo, con l'aggiunta di due ulteriori integrati (*opzionali*), è possibile avere anche un segnale a 10Hz ed uno ad 1Hz. Si parte, come detto, da un quarzo a 4.9152MHz collegato ad un CD4060B, internamente composto da un oscillatore e da 14 stadi *ripple-carry binary counter* il cui schema, recuperabile dal datasheet, è quello proposto nella **Fig. 1**. In esso ogni singolo blocchetto identificato con la sigla Ffxx, dei 14 esistenti nell'integrato, è struttu-

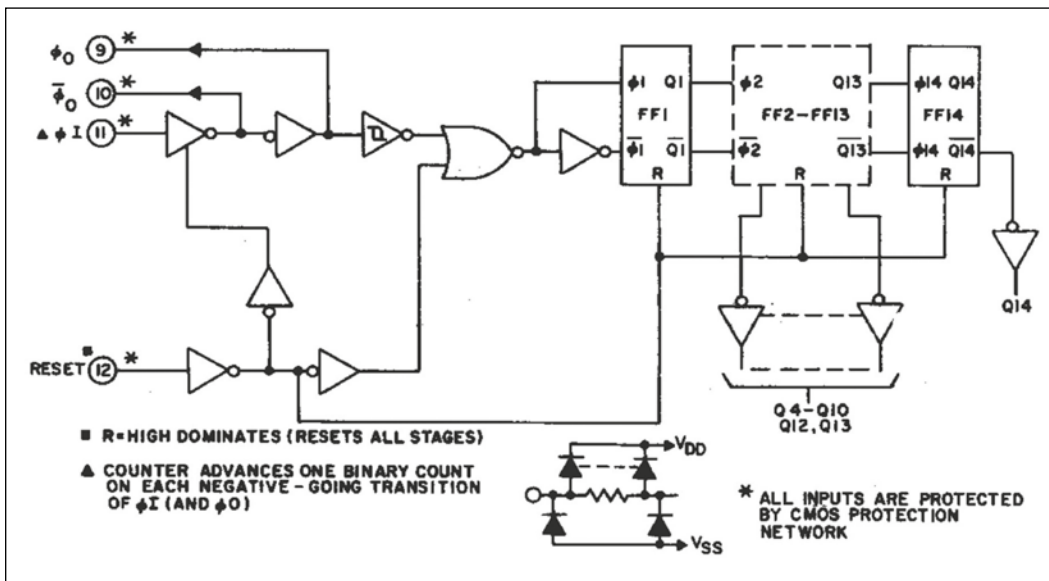


Fig. 1
Schema
funzionale
del CD4060.

rato logicamente come proposto nella **Fig. 2** ed è sostanzialmente un flip-flop. Prelevando il segnale dall'uscita Q12 si ottiene (come segnale d'uscita) un'onda quadra con frequenza pari a quella di ingresso (clock) divisa per 4.096 ovvero, partendo dai 4.9152MHz dell'oscillatore quarzato, su Q12 avremo un segnale alla frequenza di 1.200Hz. Al CD4060B faremo seguire un altro integrato estremamente comune, il 74LS92N che, altro non è se non un divisore per 12 il quale, prendendo in ingresso il segnale dall'uscita Q12 del CD4060B, genera sull'uscita un segnale a 100Hz, ovvero proprio la frequenza che ci interessava avere. A questo punto, opzionalmente (nello schema elettrico sono stati inseriti), è possibile aggiungere due contatori decimali CD4017B, che utilizziamo qui come divisori di frequenza per dividere ulteriormente il nostro segnale, una prima volta per ottenere 10Hz, ed una seconda volta per ottenere 1Hz, nel caso si necessitasse avere anche queste frequenze.

Lo schema del circuito che andremo a realizzare è quello proposto in queste pagine, dove troviamo gli integrati IC1 e IC2 a realizzare il blocco generatore a 100 Hz e IC3 e IC4 impiegati per implementare le divisioni per 10 e per 100.

La tensione di alimentazione è di 5V; il quarzo utilizzato nel circuito per generare il clock richiede i due soliti condensatori da 20/22 pF verso massa e, ai suoi estremi, una resistenza di carico da 1 MΩ.

GENERATORE CON MCU

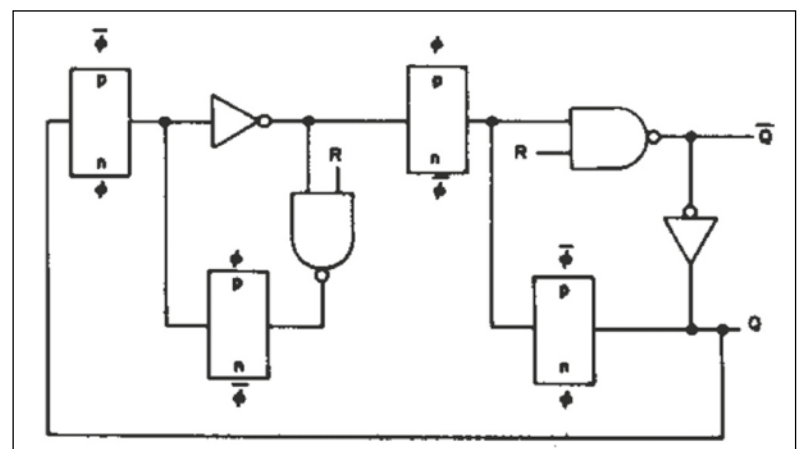
La seconda soluzione, di cui abbiamo accennato,

per generare le stesse tre frequenze è invece basata su un semplicissimo circuito utilizzando un microcontrollore a 8 bit ATtiny24 di Microchip/Atmel, che opererà con l'oscillatore interno sincronizzato da un quarzo ad 8MHz.

Si è optato per la versione "maggiorata" a 14 pin, ATtiny24, rispetto alla versione "ridotta" ad 8 pin, ATtiny25, perché, quest'ultimo non ha un timer a 16 bit che invece ci occorre per ottenere esattamente i 100Hz a partire dal quarzo a 8 MHz. Naturalmente, se già li si ha in casa, è possibile utilizzare anche il ATtiny44 o ATtiny84, che hanno solo più memoria Flash rispetto alla versione 24 che, comunque, per il nostro scopo è già più che sufficiente. Lo schema a blocchi di tale MCU è visibile nella **Fig. 3**.

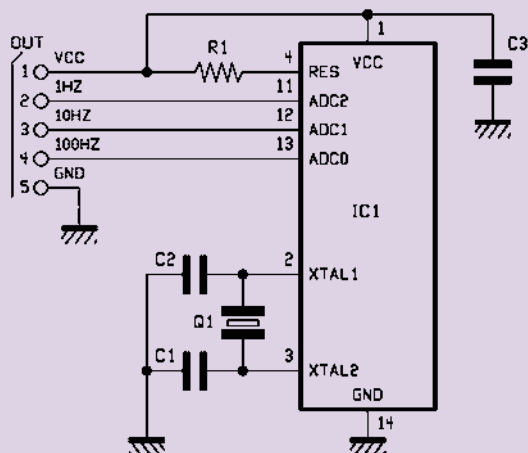
Nell'ATtiny24 che abbiamo scelto, il Timer 1 è un

Fig. 2
Costituzione del
blocco FF.

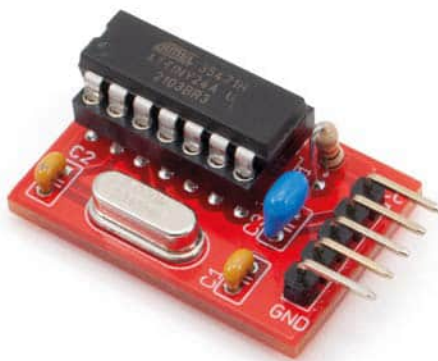


schema elettrico

GENERATORE A MCU

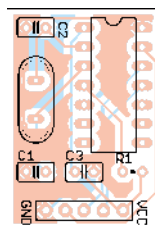


➔ Prototipo del generatore con microcontrollore.



piano di montaggio

GENERATORE A MCU



Elenco Componenti:

R1: 10 kohm
C1, C2: 20 pF
C3: 100 nF
Q1: 8 MHz
IC1: ATTINY24/44/84-PU

Varie

- strip maschio 5 poli 90°
- stampato S1571 (30x21 mm)

timer a 16 bit, mentre la memoria Flash è di 2 kbyte e la memoria SRAM di 128 Byte. Scelto Timer 1 per generare l'interrupt necessario, abbiamo scelto i pin PA0, PA1 e PA2 come uscite rispettivamente a 100, 10 ed 1Hz. Il relativo schema elettrico è proposto in queste pagine. Interessante notare che l'ATtiny24 può operare con frequenze comprese tra 0 e 10MHz con tensioni comprese tra 2,7 e 5,5V quindi, l'oscillatore realizzato in questo modo, può essere alimentato sia a 3,3V che a 5V per poter poi essere collegato sul pin di interrupt di schede che funzionano sia a 3,3V (ad esempio Arduino Pro Mini 3,3V) che a 5V.

REALIZZAZIONE PRATICA

Per la costruzione di entrambi i generatori abbiamo previsto un piccolo circuito stampato che raccoglie tutti i componenti; potete realizzare quello che vi occorre scaricando le relative tracce lato rame dal nostro sito www.elettronica.in.it e procedendo per fotoincisione. Notate che per facilitare la realizzazione, tutti i componenti sono a montaggio THT, quindi non richiedono particolari attrezzature o competenze specifiche per il montaggio. Per quel che riguarda il primo generatore, ossia quello a logica, i due integrati IC3 ed IC4 (nello schema) sono opzionali e servono solo per ottenere le due frequenze aggiuntive, quindi, se non occorrono, possono anche non essere montati. Quanto al secondo, il rispettivo piano di montaggio è in questa pagina. Per tutti e due ricordate di prestare attenzione al verso di montaggio degli integrati, desumendolo dai piani di montaggio; rispettate altresì la polarità dell'alimentazione, pena il danneggiamento dei circuiti.

Per la programmazione dell'ATtiny del generatore a MCU abbiamo scelto di utilizzare il "core" di Spence Konde "ATTinyCore". Per la sua installazione occorre aggiungere, nelle preferenze dell'IDE di Arduino, la seguente riga tra quelle delle URL per il Board Manager:

http://drazzy.com/package_drazzy.com_index.json

Dopo, nel menu Tools dell'IDE, occorrerà selezionare la voce Get Board Info (**Fig. 4**). Scegliendo come Chip quello che si ha a disposizione tra ATtiny24/44/84 (nella figura si è usato un ATtiny84). Per la programmazione della MCU occorrerà avere un programmatore ISP (es: AVRISP MKII, AVR Dragon, Atmel ICE, Pololu USB AVR 3 in 1 - V2.1 o, altro programmatore adatto per chip AVR) ed una breadboard dove poter fare i collegamenti neces-

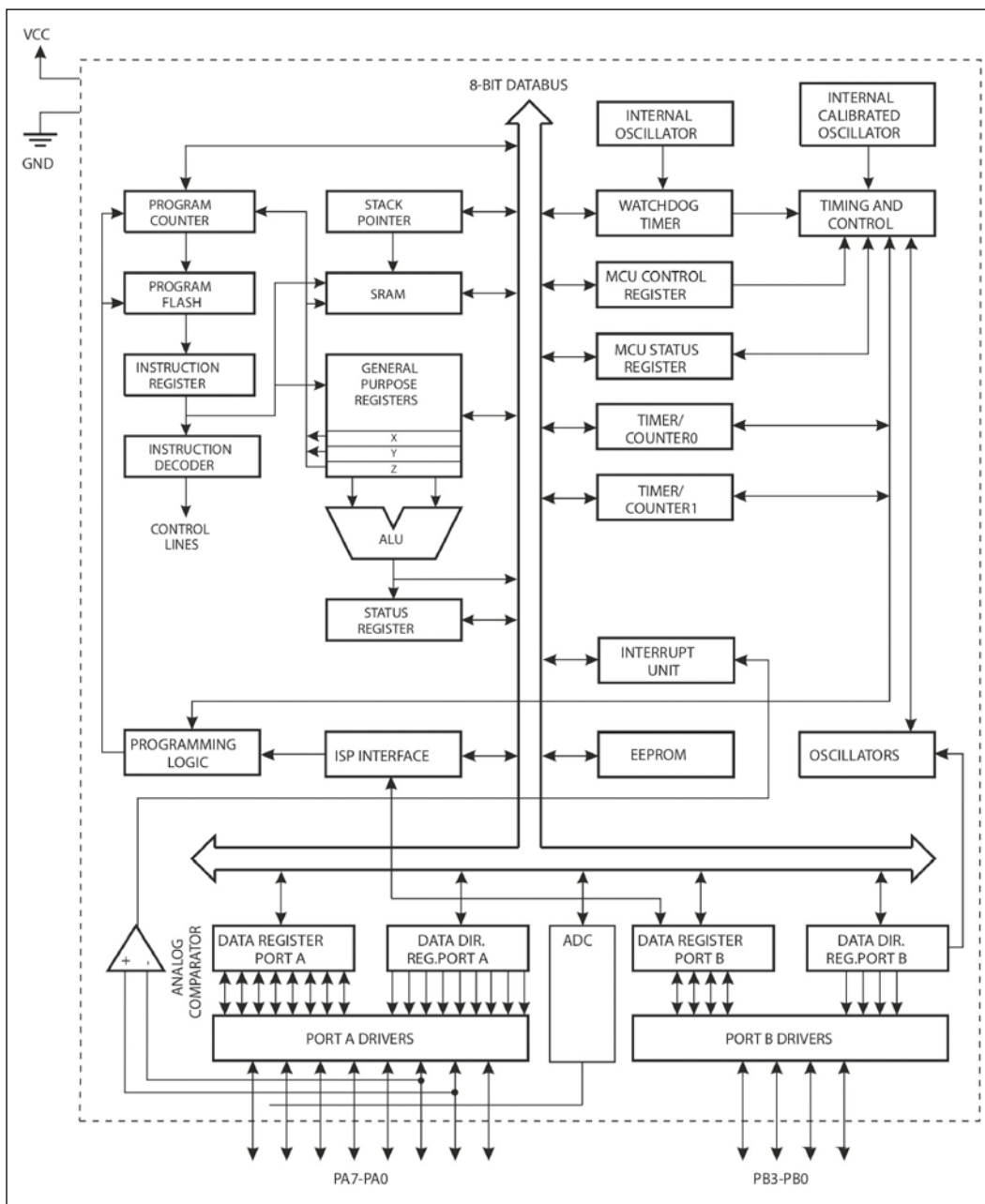


Fig. 3
Schema
a blocchi
dell'ATtiny24.

sari alla programmazione ISP. Dato che modificheremo anche i FUSE attivando il quarzo esterno, occorrerà che sulla breadboard, oltre alla MCU ed al connettore ISP per la programmazione, sia presente e connesso all'Attiny, anche il quarzo con i suoi condensatori o, una successiva riprogrammazione, non sarà possibile (la MCU, una volta cambiati i FUSE, si aspetta un quarzo collegato). Come prima prova, per verificare il corretto caricamento e funzionamento del programma, programmeremo i FUSE per lavorare con l'oscillatore interno a 8MHz (così il quarzo non sarà necessario

ed avremo comunque i tre segnali in uscita anche se non molto precisi causa la poca precisione dell'oscillatore interno). Solo dopo aver verificato il corretto funzionamento del codice, modificheremo i FUSE per cercare il quarzo esterno (che, ribadiamo, da questo momento diventerà obbligatorio). Quindi, come prima cosa, andremo a programmare i FUSE in modo da avere l'oscillatore interno attivo ad 8MHz. I valori da impostare saranno quindi:

LOW: 0xE2
HIGH: 0xDF



▼ Listato 1

```
#define PIN_100HZ 10
#define PIN_10HZ 9
#define PIN_1HZ 8

volatile uint8_t pin100Hz = 0;
volatile uint8_t pin10Hz = 0;
volatile uint8_t cnt10Hz = 0;
volatile uint8_t pin1Hz = 0;
volatile uint8_t cnt1Hz = 0;

ISR ( TIMER1_COMPA_vect ) {
    pin100Hz ^= 1;
    digitalWrite ( PIN_100HZ, pin100Hz );
    //
    cnt10Hz++;
    if ( cnt10Hz == 10 ) {
        cnt10Hz = 0;
        pin10Hz ^= 1;
        digitalWrite ( PIN_10HZ, pin10Hz );
    }
    //
    cnt1Hz++;
    if ( cnt1Hz == 100 ) {
        cnt1Hz = 0;
        pin1Hz ^= 1;
        digitalWrite ( PIN_1HZ, pin1Hz );
    }
}

void initTimer ( void ) {
    noInterrupts();
    //
    // Clear timers registers
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;
    //
    // Init Timer1 (16 bit) to generate 200Hz in-
    // terrupt necessary to
    // have 100Hz square wave from 8MHz clock
    // (8000000/((624+1)*64)) = 200
    OCR1A = 625;
    TCCR1B |= ( 1 << WGM12 );
    TCCR1B |= ( 1 << CS11 ) | ( 1 << CS10 );
    TIMSK1 |= ( 1 << OCIE1A );
    //
    //
    interrupts();
}

void setup() {
    //
    // initialize unused pins as INPUT_PULLUP
    for ( uint8_t p = 2; p < 8; p++ ) {
        pinMode ( p, INPUT_PULLUP );
    }
    //
    // initialize OUTPUT pins
    pinMode ( PIN_100HZ, OUTPUT );
    pinMode ( PIN_10HZ, OUTPUT );
    pinMode ( PIN_1HZ, OUTPUT );
    digitalWrite ( PIN_100HZ, LOW );
    digitalWrite ( PIN_10HZ, LOW );
    digitalWrite ( PIN_1HZ, LOW );
    //
    // start the timer1
    initTimer();
}

void loop() {
}
```

EXT: 0xFF

Dopo di che dovremo scrivere nell'IDE, compilare e caricare attraverso il programmatore il codice riportato nel **Listato 1**.

Provato il circuito e verificato che effettivamente le tre forme d'onda siano presenti sui pin del ATtiny24/44/84, si può passare a modificare i fuse in modo che venga utilizzato il quarzo esterno:

LOW: 0xFF

HIGH: 0xDF

EXT: 0xFF

Da questo momento, sia per leggere/scrivere i FUSE che per riprogrammare la MCU sarà necessario il quarzo esterno necessario affinché l'oscillatore della MCU possa funzionare.

ESEMPIO D'USO CON ARDUINO

Come abbiamo accennato, gli oscillatori proposti possono essere utilizzati con Arduino, in special modo con quelli che non hanno un quarzo sulla MCU, per ottenere una temporizzazione quarzata precisa al centesimo di secondo. Abbiamo già la funzione `millis()` che ritorna il contatore gestito normalmente dal Timer 0 di Arduino, quindi dovremo come prima cosa creare una ISR (*Interrupt Service Routine*) che gestisca un contatore, dopo di che, occorrerà creare una nuova funzione, ad esempio di nome `cents()`, che ritorni il contatore della nostra ISR. Inizieremo il nostro programma su Arduino con una `#include` di alcuni moduli definiti nella libreria AVR libc che, come è noto, è sempre inclusa quando si compila per Arduino AVR e poi dichiareremo il nostro contatore dei centesimi di secondo come variabile a 32 bit con attributo "volatile" (dato che la manipoliamo da dentro una ISR). A ciò faremo seguire la nostra ISR e la funzione **`cents()`** che ritorna il valore di detto contatore. Da notare che, per evitare che il valore ritornato venga alterato, nel caso nel momento della lettura dovesse scattare l'interrupt, definiremo la parte che ritorna il valore come "atomica", ovvero non interrompibile. Definiremo poi una funzione di inizializzazione che avrà il compito di collegare il pin scelto, tra quelli che possono sollevare un interrupt (e passato come parametro alla funzione), alla nostra ISR ritornando un valore booleano per indicare il successo o il fallimento dell'operazione (**Listato 2**). A questo punto abbiamo tutto ciò che ci occorre per poter scrivere una nostra versione del "blink" che però, invece di usare la funzione **`delay()`** o

```

Board: "ATtiny24/44/84(a) (No bootloader)"
Chip: "ATtiny84(a)"
Clock Source (Only set on bootloader): "8 MHz (internal)"
Pin Mapping: "Counterclockwise (like old ATTinyCore and x41-seri..."
LTO (1.6.11+ only): "Enabled"
tinyNeoPixel port: "Port A (CW:0~7,CCW:3~10)"
millis()/micros(): "Disabled (saves flash)"
Save EEPROM (only set on bootloader): "EEPROM retained"
B.O.D. Level (Only set on bootloader): "B.O.D. Disabled (saves power)"
Port
Get Board Info
Programmer: "AVR Dragon ISP mode (ATTinyCore)"
Burn Bootloader

```

Fig. 4
Impostazioni
dell'IDE per la
programmazione
della MCU.

Listato 2

```

#include <util/atomic.h> // modulo presente in AVR libc
volatile uint32_t centsCnt = 0;
// variabile contatore dei centesimi di sec.
void centsISR ( void ) { // ISR
    centsCnt++;
}

// -----
uint32_t cents ( void ) { // funzione che ritorna il contatore
    uint32_t centsTmp;
    ATOMIC_BLOCK ( ATOMIC_RESTORESTATE ) {
        centsTmp = centsCnt;
    }
    return centsTmp;
}

// -----
uint8_t centsBegin ( uint8_t pinInt ) {
    // funzione di inizializzazione della ISR
    int8_t intNum;

    pinMode ( pinInt, INPUT_PULLUP );
    intNum = digitalPinToInterrupt ( pinInt );
    if ( intNum < 0 ) return 0;

    attachInterrupt ( intNum, centsISR, FALLING );
    return 1;
}

```

Listato 3

```

uint8_t ledStatus = 0;
uint32_t lastCents = 0;

void setup() {
    pinMode( LED_BUILTIN, OUTPUT );
    digitalWrite( LED_BUILTIN, ledStatus );

    // si chiama la funzione di init per collegare il pin fisico
    // alla ISR

    if ( !centsBegin ( 2 ) ) {
        while ( true ) {
            ledStatus = !ledStatus;
            digitalWrite ( LED_BUILTIN, ledStatus );
            delay ( 100 );
        }
    }
}

void loop() {
    // facciamo lampeggiare il led un secondo ON ed uno OFF

    if ( cents() - lastCents >= 100 ) {
        ledStatus = !ledStatus;
        digitalWrite ( LED_BUILTIN, ledStatus );
        lastCents = cents();
    }
}

```

millis(), usa **cents()**. Immaginando di aver collegato il generatore quarzato a 100Hz al pin 2 di un Arduino UNO, dovremo aggiungere, di seguito a quanto scritto nel **Listato 2**, il semplice codice proposto nel **Listato 3**, in modo da avere un unico programma compilabile. Una volta compilato e caricato il codice, vedremo il LED lampeggiare esattamente con un intervallo di un secondo tra ogni ON e OFF. Naturalmente questo è solo un semplice esempio, però fornisce tutte le informazioni di come utilizzare il generatore con quei modelli di Arduino non dotati di quarzo ed avere una misura del tempo precisa al centesimo di secondo. □



Cosa occorre?

I componenti utilizzati in questo progetto sono di uso comune e facilmente reperibili. Il PCB può essere realizzato utilizzando il nuovo servizio di stampa circuiti stampati (cod. PCBPRODUCTION) direttamente dal sito www.futurashop.it. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

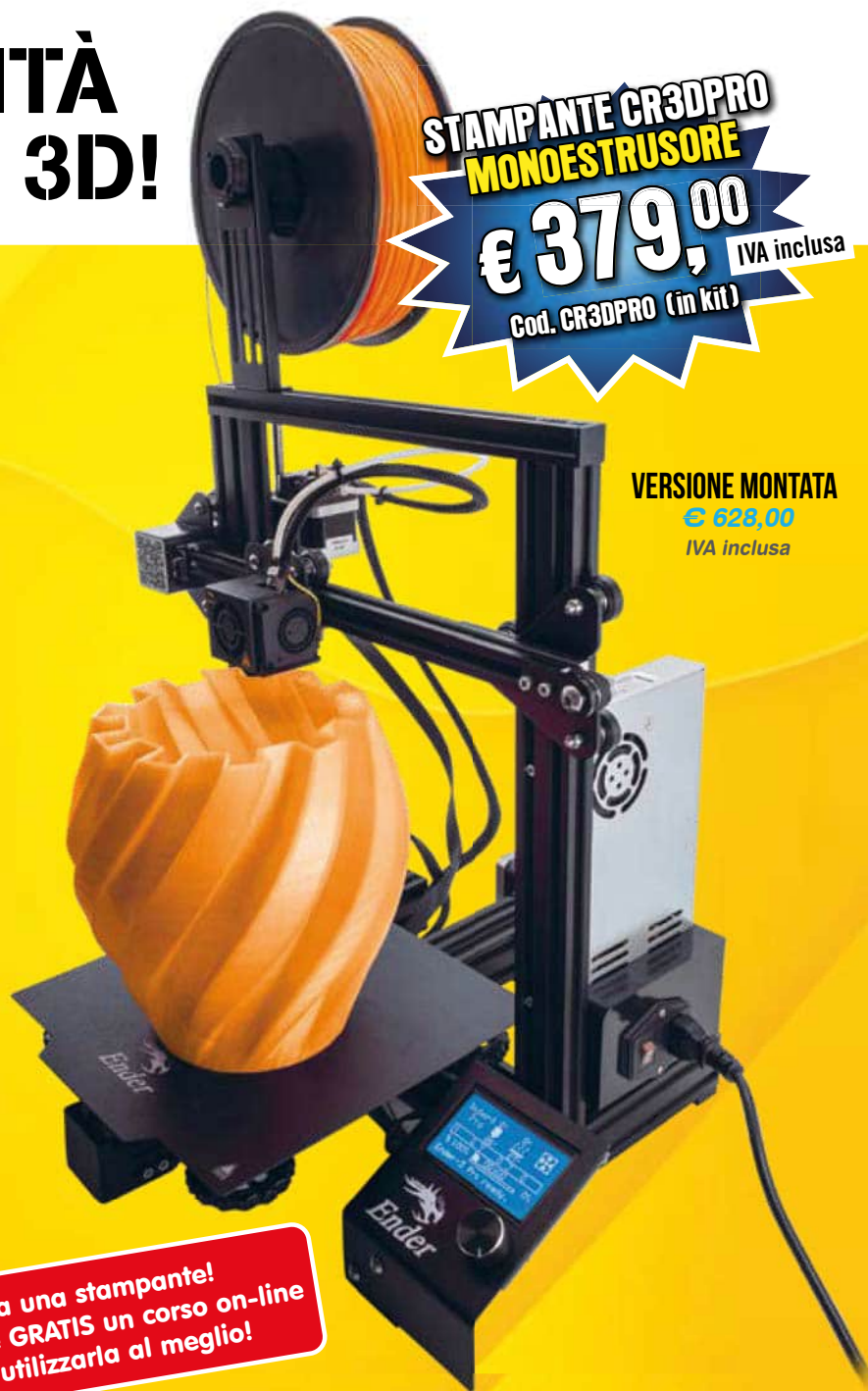
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - <http://www.futurashop.it>

ULTIME NOVITÀ DAL MONDO 3D!

CR3DPRO

**STAMPANTE 3D PER STAMPE
FINO A 22 x 22 x 25 cm**

- ✓ Tecnologia FFF FDM
- ✓ Diametro filamento 1,75 mm
- ✓ Volume di stampa (xyz) 220 x 220 x 250 mm
- ✓ Dimensione Nozzle 0,4 mm
- ✓ Estrusore MK10
- ✓ 4 motori passo-passo Nema17 (200 step)
- ✓ Spostamento asse Z con vite trapezia Ø8 mm a 3 principi in acciaio INOX
- ✓ Struttura con profilati in alluminio
- ✓ Scocca in alluminio a struttura aperta
- ✓ Livellamento del piatto manuale di precisione
- ✓ Schermo LCD
- ✓ Firmware Marlin open source
- ✓ Connettività SD - USB
- ✓ Sistema di recupero della stampa
- ✓ Slicing consigliato Cura
- ✓ Sistemi operativi Windows, Mac OSX, Linux
- ✓ Dimensioni telaio (xyz) 440 x 410 x 650 mm
- ✓ Peso 5,6 Kg
- ✓ Alimentazione 200-240 V / 3.4 A
- ✓ Output 24VDC - 270W



**STAMPANTE CR3DPRO
MONOESTRUSORE**
€ 379,00 IVA inclusa
Cod. CR3DPRO (in kit)

VERSIONE MONTATA
€ 628,00
IVA inclusa

Acquista una stampante!
Potrai seguire GRATIS un corso on-line
su come utilizzarla al meglio!



VERSIONE MONTATA
€ 1148,00
IVA inclusa

CRCP3IN1

**MACCHINA 3 IN 1:
STAMPANTE 3D,
INCISORE LASER
E FRESATRICE CNC**

**CRCP3IN1
MACCHINA 3 IN 1**
€ 899,00 IVA inclusa
Cod. CRCP3IN1 (in kit)

- ✓ Telaio in profilati di alluminio V-slot da 40 x 40 mm
- ✓ Scocca in alluminio a struttura aperta
- ✓ 5 motori passo-passo Nema17 (200 step)
- ✓ Spostamento asse Z: doppia vite trapezia Ø8 mm a 3 principi in acciaio INOX
- ✓ Struttura con profilati in alluminio 40 x 40 mm
- ✓ Schermo grafico touch da 4,3"
- ✓ Firmware Marlin open source
- ✓ Connettività SD - USB
- ✓ Alimentazione 200-240 V / 3.4 A
- ✓ Output 24 VDC - 270 W
- ✓ Dimensioni telaio (xyz): 400 x 450 x 650 mm
- ✓ Peso 10 Kg

PIATTO PER SALDATURA A RIFUSIONE



di **MASSIMO DIVITO**

Soluzione semplice ed economica per assemblare o riparare piccoli circuiti SMD nel nostro laboratorio.

A

Il giorno d'oggi l'utilizzo di componenti SMD anche da parte degli hobbisti è sempre più frequente, vuoi per la miniaturizzazione dei circuiti, vuoi perché alcuni componenti si trovano solo in questo formato.

Ecco perché tutti noi, prima o poi, saremo alle prese con la saldatura di un

componente per montaggio superficiale.

Rispetto agli elementi per montaggio a foro passante (THT=Through Hole Technique), armeggiare con i componenti SMT (Surface Mount Technique) o SMD che dir si voglia, richiede una buona vista, una mano decisamente ferma, un saldatore dotato di una punta adeguata e, per finire, ma forse è la cosa più importante, una buona dose di pazienza. Quando però la presenza in un circuito di componenti a montaggio superficiale è prevalente e soprattutto quando si tratta di chip con molti piedini, la situazione si complica, tanto più se

vi sono molti elementi ravvicinati; in questo caso utilizzare il saldatore può diventare improponibile. Per tale motivo abbiamo pensato ad un progetto che potesse venire incontro alle vostre esigenze in situazioni del genere: qualcosa di semplice e poco dispendioso che fosse facilmente realizzabile. La tecnica di saldatura che si utilizza in ambito professionale per gli SMD viene chiamata reflow soldering, in italiano saldatura per rifusione, ed utilizza normalmente dei forni; invece noi abbiamo pensato a un piano riscaldato, adatto a circuiti più piccoli, che occupa poco spazio nel nostro laboratorio e che all'occorrenza può essere utilizzato come preriscaldatore per le operazioni di saldatura e dissaldatura con la stazione ad aria calda. La saldatura per rifusione è un processo in cui viene utilizzata una pasta saldante composta da stagno in polvere e flussante liquido anziché la normale lega di stagno/piombo (o lead-free...) per saldare i componenti sul circuito stampato. Sulle piazzole del circuito stampato viene depositata la pasta saldante tramite una siringa oppure tramite un apposito stencil, dopo di che vengono prima posizionati tutti i componenti elettronici, ed infine l'intero assemblaggio viene sottoposto ad un trattamento termico controllato, che fonde la lega e salda permanentemente i componenti.

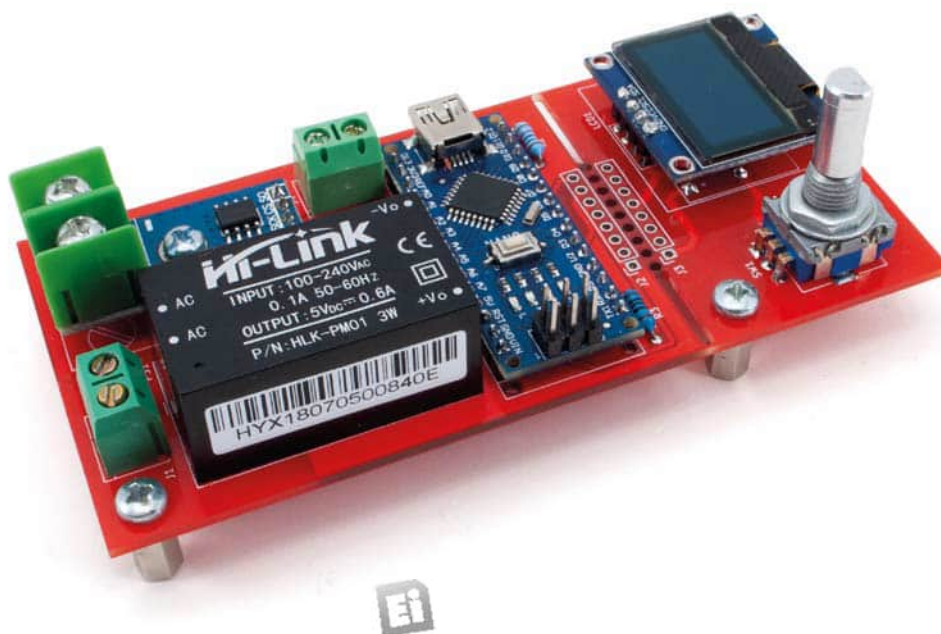
COME FUNZIONA

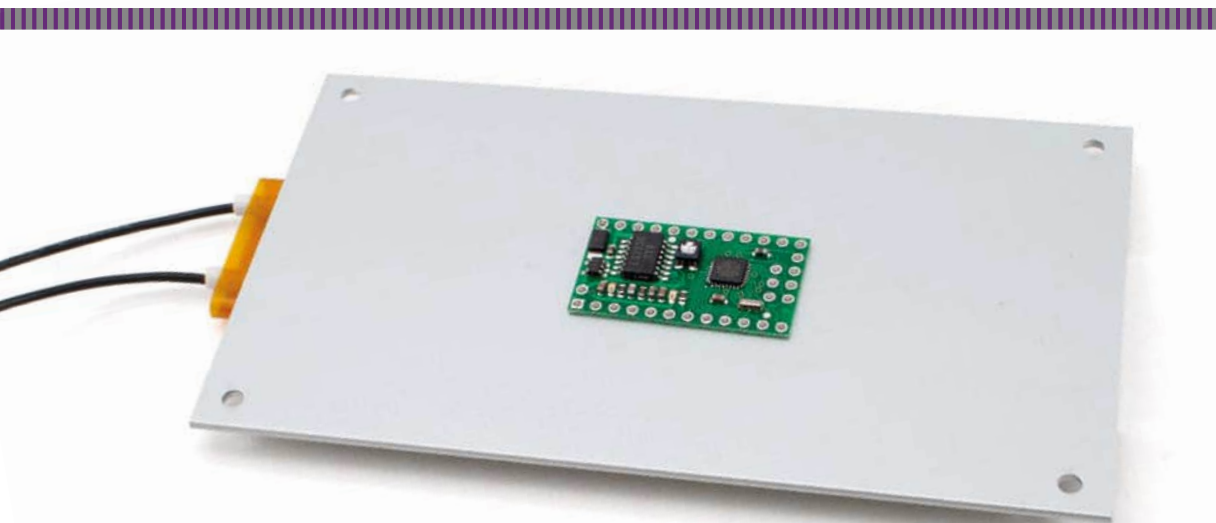
Il nostro progetto si compone di un piano in alluminio contenente delle resistenze elettriche funzionanti a 230 volt che lo riscaldano; su questo piano viene appoggiato il circuito stampato da saldare. Le resistenze vengono gestite tramite un relé a stato solido che viene comandato a sua volta da

un circuito basato su Arduino Nano. Al circuito è collegata una termocoppia di tipo K con anello di fissaggio, applicata a vite al piano, che ne legge costantemente la temperatura. Il tutto viene gestito tramite un encoder rotativo che permette di selezionare la modalità di funzionamento tra due diversi profili: uno per le leghe di stagno con il piombo ed uno per le leghe senza piombo. La terza modalità permette di regolare il piano ad una temperatura fissa, tale funzionalità è utile ad esempio per preriscaldare un circuito sul quale andremo a lavorare con uno stilo ad aria calda. Abbiamo poi aggiunto un piccolo display OLED da 0,96 pollici sul quale andremo a selezionare il profilo che vogliamo avviare e, durante le fasi di rifusione, ci mostrerà la temperatura impostata e quella raggiunta dal piano oltre, al tempo mancante al completamento del processo. Come abbiamo già spiegato, per effettuare una corretta saldatura è necessario seguire una precisa curva di temperatura per permettere alla pasta saldante di fondersi correttamente ed allo stesso momento evitare che i componenti si danneggino per il troppo calore; di norma ogni costruttore specifica, per i propri integrati, la curva ottimale. L'intero processo è suddiviso in quattro fasi:

- Preheat: gli aggreganti della pasta saldante iniziano ad evaporare;
- Soak: gli aggreganti completano l'evaporazione, il flussante si scioglie e deossida le piazzole;
- Reflow: la parte metallica della lega si fonde e crea una connessione meccanica ed elettrica permanente tra i componenti ed il circuito stampato; il flussante evapora;
- Cooling: lo stagno si raffredda solidificandosi.

→ Il nostro prototipo pronto per essere inserito nel contenitore.





← Per la saldatura è possibile appoggiare la basetta al piano riscaldato, dopo aver cosparso le piazzole con la pasta saldante.

Per fare questo è necessario conoscere ad ogni istante del processo di rifusione la temperatura che dovrebbe avere il piano in quel determinato momento.

Dato che la temperatura del piano viene letta all'incirca ogni 250 millisecondi, per un singolo profilo della durata di 380 secondi dobbiamo conoscere 152 valori di temperatura ($380.000/250$). Memorizzare ed accedere ad un numero così elevato di valori diventa una cosa abbastanza complicata a livello firmware ed eventuali future modifiche risulterebbero macchinose, considerando inoltre che nel nostro caso i profili sono due (ma volendo ne potremmo aggiungere altri) il numero di valori si raddoppia.

Ecco perché siamo arrivati ad una soluzione diversa grazie alla quale possiamo definire solo alcune delle temperature e tramite esse calcolare le altre. Se guardiamo i grafici dei profili, visibili nella **Fig. 1**, notiamo che le variazioni di temperatura, sia in salita che in discesa, sono lineari.

Possiamo quindi suddividere ogni profilo in diversi segmenti dei quali dobbiamo conoscere soltanto gli estremi.

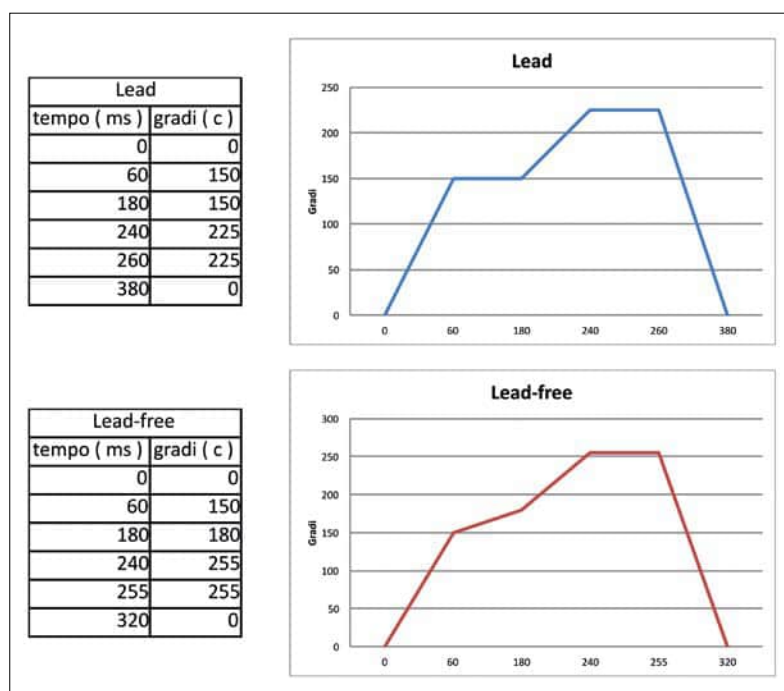
Ogni segmento è definito da due coppie di valori che ne indicano inizio e fine ed ogni coppia è composta dai valori, rispettivamente dei secondi e della temperatura; tali valori sono facilmente memorizzabili in un array bidimensionale composto da sei elementi: i sei vertici che descrivono appunto il profilo. In questo modo, volendo conoscere il valore di temperatura da impostare sul piano ad un determinato tempo non dovremo far altro che applicare il metodo dell'interpolazione lineare che permette di trovare un valore sconosciuto compreso tra due valori noti.

Conoscendo il valore di temperatura che il piano dovrebbe avere ed il valore che realmente ha possiamo regolarne di conseguenza l'accensione o lo spegnimento.

Per fare in modo che la temperatura segua con maggiore precisione quella indicata dal profilo utilizzeremo la tecnica del controllo PID. Tale tipo di controllo non fa altro che regolare una grandezza monitorandola costantemente e agendo su alcune variabili per cercare di tenerla prossima al valore impostato. Chiaramente il funzionamento non è così banale anzi segue delle precise formule matematiche, per fortuna per Arduino esistono apposite librerie che ci semplificano di molto la vita.

Come abbiamo già detto utilizziamo per l'accensione del piano un relé a stato solido che ha la caratteristica di avere al suo interno una funzionalità chiamata Zero Cross Trigger Method che fa sì che il cambio di stato della sua uscita, vale a dire il

Fig. 1
Profili per la saldatura con l'utilizzo di leghe con e senza stagno.



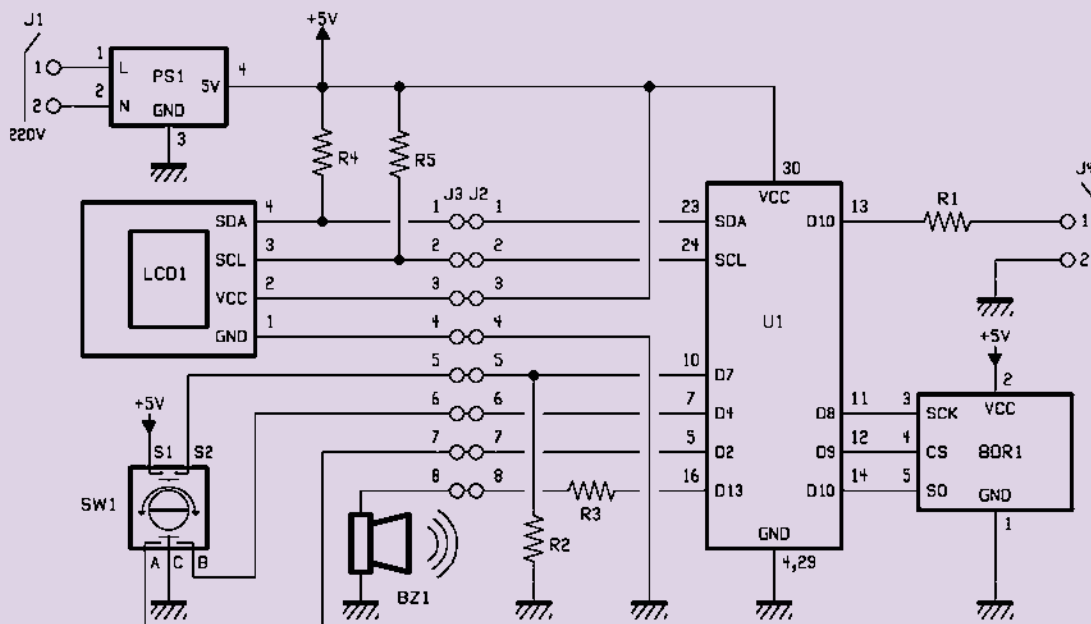
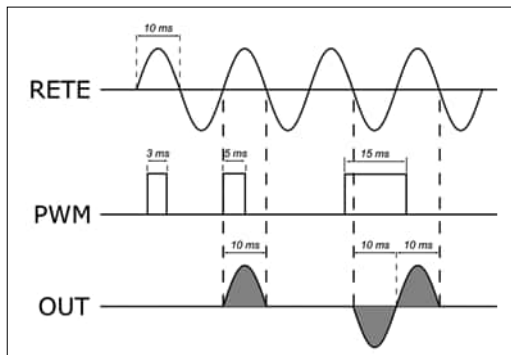
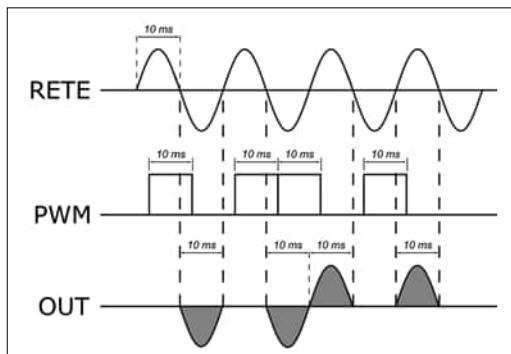


Fig. 2
Principio di
funzionamento
dello Zero Cross
Trigger Method
pilotato in PWM.



passaggio tra ON ed OFF e viceversa, avviene solo quando l'onda della tensione di rete, che ricordiamo essere a 50Hz, passa per lo zero. Questo significa che da quando noi diamo o togliamo tensione all'ingresso del relé a quando

Fig. 3
Comportamento
Zero Cross Trigger
Method pilotato in
PWM a 1 Hz.



lo stesso accende o spegne il carico, potrebbe passare qualche millisecondo. Chiaramente questa caratteristica non ci consente di avere un controllo esatto dell'uscita, difatti se noi andiamo a pilotare il relé con un impulso della durata inferiore alla durata di una semionda, che magari inviamo un millisecondo dopo il passaggio dallo zero abbiamo come risultato la mancata accensione del carico. Se invece inviamo l'impulso all'esatto passaggio dallo zero, abbiamo sì l'accensione del carico ma il suo spegnimento si ha solo al prossimo passaggio per lo zero. Prendendo come esempio la **Fig. 2** a fronte di un PWM che ha pilotato il relé per un totale di 23 millisecondi ci ritroviamo l'uscita attiva per un totale di 30 millisecondi.

Una soluzione sarebbe quella di utilizzare un circuito Zero Crossing Detector che, come fa capire il nome, è in grado di rilevare tutte le volte che la tensione di rete passa per lo zero in modo da sincronizzare correttamente l'uscita PWM.

Nel nostro caso, per non complicare ulteriormente il circuito, abbiamo risolto questo problema utilizzando una frequenza di PWM molto bassa e più precisamente di 1Hz, in questo modo avremo un periodo totale di 1 secondo, o meglio 1000 millisecondi, suddiviso in 100 step da 10 millisecondi ciascuno. In pratica per ogni variazione del duty-cycle di un punto percentuale avremo una variazione della lunghezza dell'impulso del valore di 10 millisecondi che è esattamente la durata di una semionda delle frequenze di rete.

Le termocoppie sono dei trasduttori di temperatura, comunemente utilizzati in molteplici settori: industriale, scientifico, ingegneristico ecc. Grazie alle loro ridotte dimensioni ed ai tempi di risposta rapidi, possono essere impiegate anche in ambienti particolarmente critici, riuscendo a misurare in modo accurato un range esteso di temperature. Il loro grosso vantaggio è che sono molto semplici, resistenti ed economiche.

Le termocoppie sono progettate per la misurazione delle alte temperature di corpi fisici, componenti e sostanze.

Ogni termocoppia è formata da due parti di metallo differente, uniti in un punto chiamato "giunto caldo"; questo punto viene messo a contatto con l'oggetto o la sostanza della quale si vuole misurare la temperatura. L'altra estremità, quella formata dai due fili liberi, è chiamata "giunto freddo" e se viene collegata ad un voltmetro, la differenza di temperatura tra i due giunti crea una differenza di potenziale proporzionale alla temperatura (effetto Seebeck). Le termocoppie possono avere un aspetto esteriore anche molto diverso tra loro ma alla base tutte hanno lo stesso principio di funzionamento. Il giunto freddo è origina-

to dalla differenza tra il metallo dei fili della termocoppia e quello dei cavi di collegamento.

Vi sono diversi tipi di termocoppie ma quella di tipo K è sicuramente la più diffusa nel settore industriale. È costituita da una combinazione di fili a base di nichel e gli intervalli di misurazione sono compresi tra -200 e +1260 gradi centigradi, inoltre la resistenza alla corrosione e ossidazione del nichel ne permette l'impiego in una vasta gamma di applicazioni.

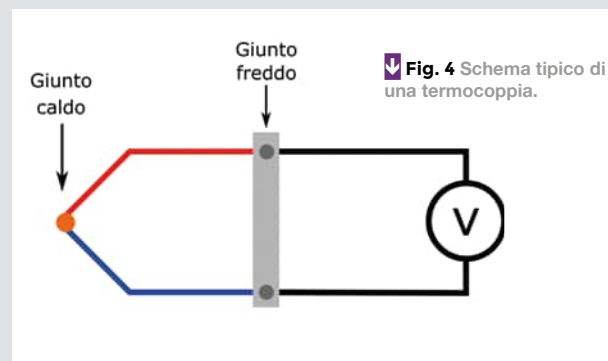


Fig. 4 Schema tipico di una termocoppia.

In questo modo il valore di duty-cycle che impostiamo sarà correttamente riportato all'uscita del relé poiché abbiamo la certezza che lo stesso ritardo applicato nell'attivare il carico sarà applicato anche alla sua disattivazione, la sola differenza sarà un possibile ritardo che va da 1 a 9 millisecondi che traslerà in avanti l'intera onda (Fig. 3).

Il progetto che vi presentiamo ha solo un sistema per regolare il riscaldamento, mentre il raffreddamento avviene in modo naturale per dispersione del calore; per questo motivo la curva reale non si presenterà uguale a quella teorica.

Molto dipende dall'ambiente in cui operiamo, ma dopo svariate prove vi assicuriamo che non abbiamo mai avuto problemi a portare a termine con ottimi risultati il processo di rifusione.

SCHEMA ELETTRICO

Grazie all'utilizzo di Arduino (U1) il circuito è molto semplice; infatti oltre ad esso vi è solo un modulo con a bordo un MAX6675. Questo chip si occupa di amplificare il segnale presente sui terminali della termocoppia di tipo K, di convertirli da analogico a digitale e di restituirceli tramite interfaccia SPI. Leggerla tramite gli ingressi analogici sarebbe infatti più complicato dato che i segnali in gioco sono nell'ordine dei microvolt e per questo motivo andrebbero opportunamente amplificati.

Per questo motivo abbiamo deciso di utilizzare questo chip e più precisamente un modulo preassemblato (BRD1) che lo monta perché il MAX6675

guarda caso lo producono solo in formato SMD!!

I segnali provenienti dalla termocoppia vengono opportunamente convertiti ed inviati tramite i piedini SCK, CS ed S0 collegati a degli ingressi digitali di U1 che li processa e decide di conseguenza se aumentare o diminuire la temperatura del piano. Il piano viene alimentato tramite un relé a stato solido, nel nostro caso abbiamo utilizzato un Fotek SSR-25-DA che supporta carichi fino a 25 ampere. Il pilotaggio di questo relé avviene tramite un'uscita digitale con interposto il resistore R1 come si fa di solito per accendere un LED, infatti andiamo a pilotare un optoisolatore collocato al suo interno, nella sezione di ingresso.

La scelta di utilizzare un tale componente è dettata dal fatto che il carico ad esso applicato viene pilotato in PWM tramite cioè un treno di impulsi che accende e spegne il piano molteplici volte al secondo.

Un normale relé elettromeccanico, oltre a non avere la velocità di commutazione necessaria, non resisterebbe a lungo ed i suoi contatti si danneggerebbero irrimediabilmente per via degli archi che si formerebbero al suo interno.

Sui piedini SDA ed SCL di Arduino abbiamo collegato, aggiungendo le resistenze di pull-up R4 ed R5, il piccolo display LCD1 da circa un pollice di dimensioni basato sul chip SSD1306.

Abbiamo poi l'encoder rotativo in quadratura SW1 con i suoi piedini A e B collegati a degli ingressi digitali di U1: come potete vedere dallo schema

In matematica, per interpolazione si intende un metodo per individuare nuovi punti del piano cartesiano a partire da un insieme finito di punti dati. Essa consente quindi di ricavare, in modo approssimativo, l'andamento di una curva conoscendone solo alcuni punti.

Nell'interpolazione lineare, che è anche il metodo più semplice per interpolare, si ha bisogno di due soli punti adiacenti, individuati con (x_a, y_a) e (x_b, y_b) , unendo i quali è possibile tracciare un segmento.

Per calcolare quindi il punto (x_0, y_0) , intermedio ai punti conosciuti, utilizziamo la formula:

$$y_0 = \frac{(x_0 - x_a)(y_b - y_a)}{x_b - x_a} + y_a$$

Questa formula può essere interpretata come valutazione della media ponderata.

Facciamo un esempio riferendoci alla Fig. 5 e sostituendo alla formula i punti del primo segmento vale a dire $(0,0)$ e $(60,150)$, ipotizzando di voler conoscere il valore di y_0 dato $x_0=20$:

$$y_0 = \frac{(20-0)(150-0)}{60-0} + 0$$

$$y_0 = \frac{20 \cdot 150}{60}$$

$$y_0 = \frac{3000}{60}$$

$$y_0 = 50$$

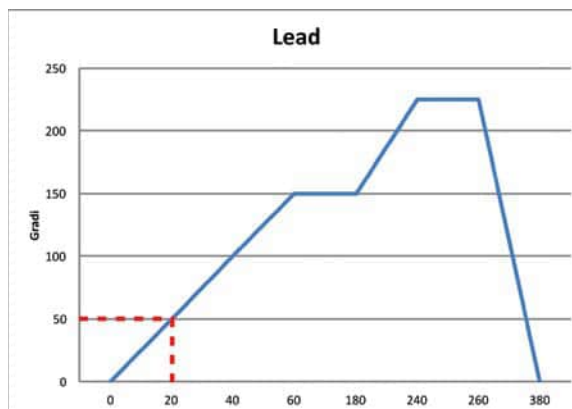


Fig. 5 - Esempio di calcolo con l'interpolazione lineare.

non abbiamo previsto dei resistori di pull-up ma abbiamo abilitato via software quelli interni al microcontrollore, il piedino C è invece collegato al GND. Il pulsante integrato all'albero dell'encoder, che permette le varie selezioni, è collegato da un lato al Vcc e dall'altro al piedino 10 di U1 insieme al resistore di pull-down R2.

Infine c'è il buzzer BZ1 collegato al piedino 9 di U1 tramite il resistore R3.

L'intero circuito è alimentato dal modulo PS1, un convertitore AC/DC della Hi-Link che si monta direttamente sul circuito stampato, questo modulo ha un ingresso a 230 volt alternati ed un'uscita a 5 volt da 600 milliampere.

Questa soluzione ci evita l'utilizzo di alimentatori esterni facilitando l'inscatolamento dell'intero circuito in un unico contenitore.

IL FIRMWARE

Lo sketch di Arduino è abbastanza corposo (lo scaricate per intero dal nostro sito www.elettronica.it) pertanto qui ne commentiamo solo alcune parti, che sono quelle meno immediate da capire o permettono di personalizzare il progetto a vostro piacimento.

Come prima cosa vengono importate le tre librerie che utilizziamo per:

- il display (U8g2 by oliver);

- il MAX6675 (MAX6675_Thermocouple by Yuri Salimo);
- il PID (PID by Brett Beauregard);
- il PWM (<https://github.com/maxint-rd/FastPwmPin>).

Vengono poi definiti i piedini collegati al MAX6675 e all'encoder rotativo oltre a due simboli grafici personalizzati che comporranno sul display l'icona del termometro.

Nelle dichiarazioni iniziali c'è da notare l'array profili che come abbiamo detto definisce i due profili di saldatura:

```
int profili [2][6][2] = {
    {{0, 0}, {60, 150}, {180, 150}, {240, 225}, {260, 225}, {380, 0}}, // profilo Lead
    {{0, 0}, {60, 150}, {180, 180}, {240, 255}, {255, 255}, {320, 0}} // profilo Lead-free
};
```

Sarà qui che dovrete intervenire nel caso vogliate modificare o aggiungere un profilo, modificando in modo opportuno i valori dell'array e ricordando che per ogni coppia di valori il primo si riferisce al tempo in millisecondi ed il secondo ai gradi; le tabelle della Fig. 1 vi renderanno più chiaro il concetto. Successivamente a questo array vi è un altro

blocco di variabili che vi potrebbe tornare utile modificare:

```
int tempMinPreheater = 60; //imposto la temperatura minima
settabile come preriscaldatore
int tempMaxPreheater = 250; //imposto la temperatura massima
settabile come preriscaldatore
int tempStepPreheater = 5; //imposto il valore degli step di
regolazione per il preriscaldatore
int tempAmbiente = 50; //imposto la temperatura ambiente
massima, se il piano è più caldo non è possibile avviare il
processo di rifusione
int tempAdjust = 0; //correzione della temperatura letta dalla
sonda
```

Sono tutte quante commentate ed è quindi facile intuire che le prime due indicano il valore minimo e quello massimo impostabile; la terza indica invece il valore che verrà aggiunto o sottratto ai gradi impostati per ogni rotazione dell'encoder, questi valori si riferiscono solo alla funzionalità *preriscaldatore*. La variabile *tempAmbiente* indica il valore massimo che può avere il piano prima di avviare il processo di rifusione ed è indicata come temperatura ambiente perché di solito il piano ha la temperatura

presente nell'ambiente in cui si trova.

Questo accorgimento evita che venga avviato un processo di saldatura subito dopo averne terminato un altro, in questo caso infatti si andrebbe a falsare la curva di rifusione con il rischio di vanificare l'intera procedura.

La variabile *tempAdjust* serve per correggere la lettura della temperatura del piano.

Il resto dello sketch non ha bisogno di commenti, la lettura dei pulsanti e dell'encoder viene fatta senza scomodare gli interrupt, vengono utilizzati alcuni timer per leggere la pressione prolungata di P1 e per aggiornare periodicamente il display.

Nel codice che gestisce la funzionalità di reflow possiamo vedere in che modo utilizziamo la formula dell'interpolazione lineare per calcolare a che temperatura impostare il piano:

```
reflowNowTime = (unsigned long)(millis() - reflowStartTime); // Calcola da quanti millisecondi è iniziato il ciclo di reflow;
for (int i = 0; i < 6; i++) {
    if ((reflowNowTime >= (profili[profilo][i][0] * secondo)) && (reflowNowTime < (profili[profilo][i + 1][0] * secondo))) {
```



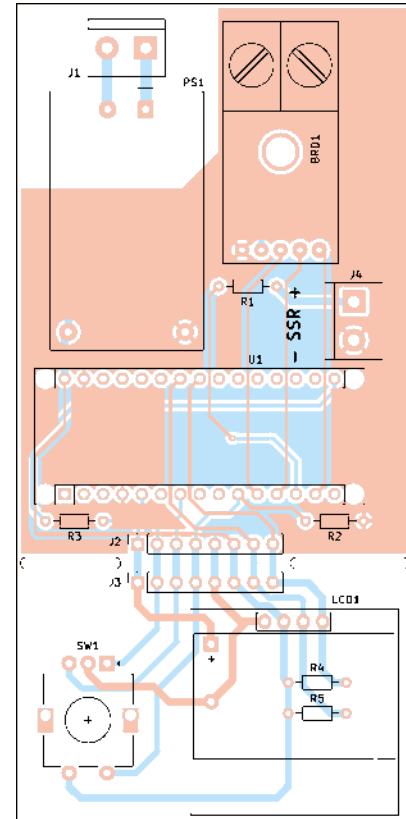
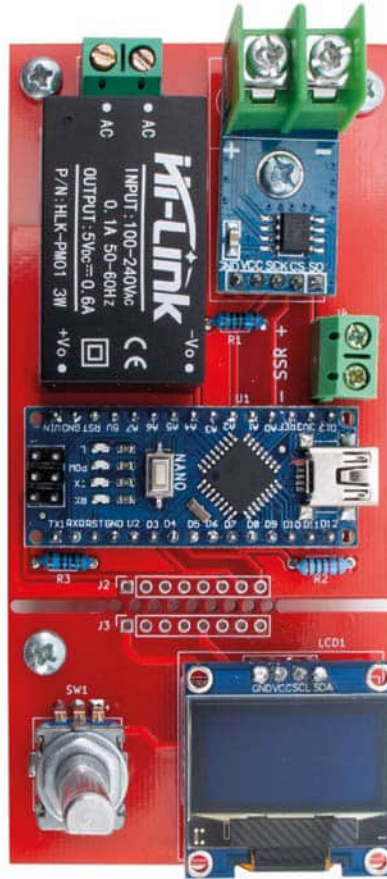
Fig. 4
Il sistema completo: la termocoppia si fissa a vite mediante il relativo occhio.

Elenco Componenti:

R1-R3: 470 ohm
 R2: 10 kohm
 R4-R5: 47 Kohm
 LCD1: OLEDGVSCSD
 (lcd oled 0,96")
 U1: Arduino Nano
 PS1: 3WMA2205V
 (modulo alimentatore)
 BRD1: MODMAX6675
 (sonda tipo K)
 BZ1: buzzer senza elettronica
 (BUZZER5V)

Varie:

- strip femmina 4 poli
- strip femmina 15 poli
- strip femmina 5 poli
- encoder rotativo
- morsetto a 2 poli
- stampato S1561 108x52



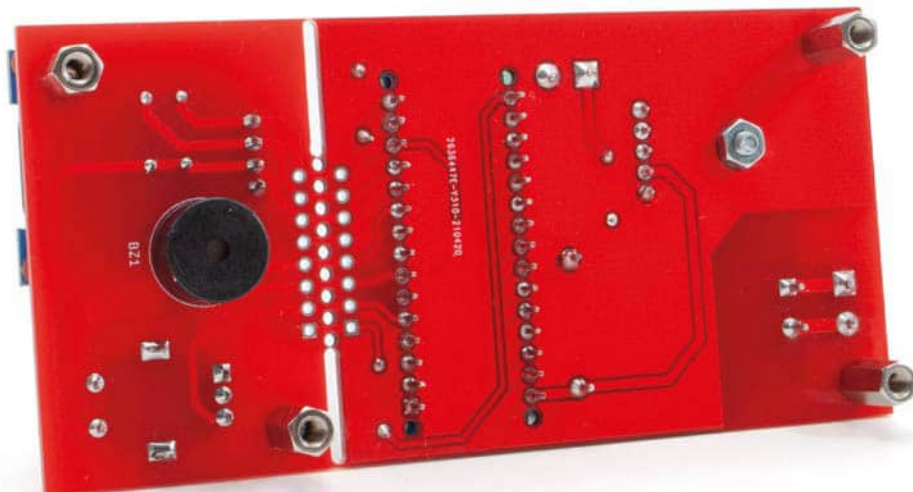
```

xa = profili[profilo][i][0] * secondo;
xb = profili[profilo][i + 1][0] * secondo;
ya = profili[profilo][i][1];
yb = profili[profilo][i + 1][1];
tempTarget = (((reflowNowTime - xa) * (yb - ya))
/ (xb - xa)) + ya;
// y0=( (x0-xa)*(yb-ya) ) / (xb-xa) + ya
// Formula interpolazione lineare
    
```

```

faseAttuale = i;
regolaRiscaldatore();
}
}
    
```

Nel ciclo *for* scorriamo l'array dei profili per ricavare i valori noti, sul piano delle *x* i millisecondi e sul piano delle *y* i gradi centigradi. In questo modo abbiamo i valori *xa*, *ya*, *xb*, *yb* mentre *x0* sarà



Il cicalino che fornisce le segnalazioni di stato.

uguale ai millisecondi passati dall'inizio della fase di rifusione: con questi valori possiamo calcolarci y_0 che rappresenta il valore della temperatura da impostare sul nostro piano riscaldante.

REALIZZAZIONE PRATICA

Per il controller è stato disegnato un apposito circuito stampato che ne facilita l'assemblaggio e permette di rendere compatto e ordinato l'insieme. Tale circuito stampato prevede che sia possibile staccare la parte dell'encoder e del display per permetterne l'installazione nelle più svariate configurazioni; nel caso decidiate di dividere il PCB dovreste ovviamente connettere le due parti tramite i connettori J2 e J3. La collocazione ottimale del circuito è in un contenitore di plastica che ne permette un totale isolamento, una soluzione potrebbe essere quella di utilizzare una cassetta di derivazione per impianti elettrici con il coperchio opportunamente forato per il display e per l'encoder rotativo. Data poi la presenza della tensione di rete, si potrebbe pensare di inscatolare insieme al circuito anche il relé allo stato solido e di far fuoriuscire la presa a 230 Volt per il collegamento del piano e la sonda a termocoppia di tipo K. Quest'ultima è importante che sia posizionata correttamente per permettere una lettura precisa della temperatura, magari ancorandola solidamente alla piastra riscaldante tramite qualche staffa metallica: una soluzione potrebbe essere quella di inserirla in un terminale ad occhiello di alluminio per poi stringerla con delle pinze senza esagerare con la pressione per evitare di danneggiarla. In questo modo la sonda (**Fig. 4**) può essere ancorata al piano con una semplice vite, aggiungendo magari alla connessione meccanica della pasta termo-conduttiva, come quella utilizzata per i processori dei PC, che favorisce la conduzione termica e migliora la lettura. Un'altra soluzione sarebbe quella di lasciarla "libera" e di farla aderire, ogni volta che si esegue un processo di rifusione, al PCB che si sta saldando, fissandola allo stesso con un pezzo di nastro kapton. Ponete estrema attenzione nella realizzazione e nel cablaggio data la presenza della tensione di rete che risulta molto pericolosa, ricordatevi di collegare il cavo di massa alla spina ed a tutte le parti metalliche.

TARATURA ED UTILIZZO

Una volta programmato Arduino ed assemblato il tutto, dopo aver controllato con attenzione i cablaggi, soprattutto quelli della 230 Volt, possiamo accendere il circuito e verificarne il corretto funzio-

namento, prima di utilizzarlo è però conveniente procedere ad una semplice ma utile taratura. In pratica, in base a dove collocheremo la sonda di tipo K, potremmo avere una lettura leggermente falsata rispetto a quella del resto piano; ciò è dovuto alla disposizione delle resistenze che sono solitamente centrate ed è proprio nella parte centrale o comunque in corrispondenza di esse, che avremo una lettura più precisa. Man mano che ci spostiamo la lettura avrà delle piccole differenze che aumenteranno all'aumentare della distanza dalla fonte di calore. Per correggere la lettura abbiamo previsto la variabile *tempAdjust* che per default è pari a 0. Se avete modo di leggere la temperatura del piano tramite un termometro ad infrarossi o di altro tipo potrete verificare la temperatura letta dal nostro circuito e quella rilevata dal termometro. Dovrete assegnare alla suddetta variabile il valore risultante dalla differenza tra le due, riuscirete in questo modo a correggere la lettura portandola ad un valore più prossimo a quello reale. Fatto ciò sarete pronti a saldare i vostri circuiti, vi consigliamo comunque di iniziare con basette di piccole dimensioni, ad esempio piccoli adattatori SOIC-DIL, dovreste infatti prendere dimestichezza con la quantità di solder-paste da depositare sulle piste ed il corretto posizionamento dei componenti.

CONCLUSIONI

Il progetto può non solo essere l'elemento riscaldante unico, ma può funzionare come preriscaldatore del PCB nella saldatura e dissaldatura con la stazione ad aria calda. In quest'ultimo caso deve portare a circa 150 °C la basetta, allorché con la hot air station si scaldano i componenti da saldare, sciogliendo flussante e lega saldante. 



Cosa occorre?

I componenti principali utilizzati in questo progetto sono acquistabili da Futura Elettronica. La board Arduino Nano (cod. ARDUINONANO) è disponibile a Euro 25,00, la piastra riscaldante (cod. 400WPTCRISC) è in vendita a Euro 18,00, il sensore con sonda K (cod. MODMAX6675) costa Euro 8,00, il display OLED (cod. OLEDGVSCSD) è in vendita a Euro 11,00, il modulo alimentatore 220VAC/5VDC (cod. 3WMA2205V) è disponibile a Euro 6,50, il buzzer (cod. BUZZER55E) è in vendita a Euro 0,50, l'encoder rotativo (cod. ENCODER12MM) costa Euro 3,50.

Il PCB può essere realizzato utilizzando il servizio di realizzazione PCB (cod. PCBPRODUCTION) disponibile sul sito [futurashop.it](http://www.futurashop.it). I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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



Il termostato diventa WiFi!

Termostato touch screen da incasso

**Istruzioni
in italiano!**

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

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

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



Prezzi IVA inclusa.

PLACCA PER TERMOSTATO WiFi FR695

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



€ 4,90

cod. **PLACCAFR695**

€ 59,00

cod. **FR695**

person: 1.0

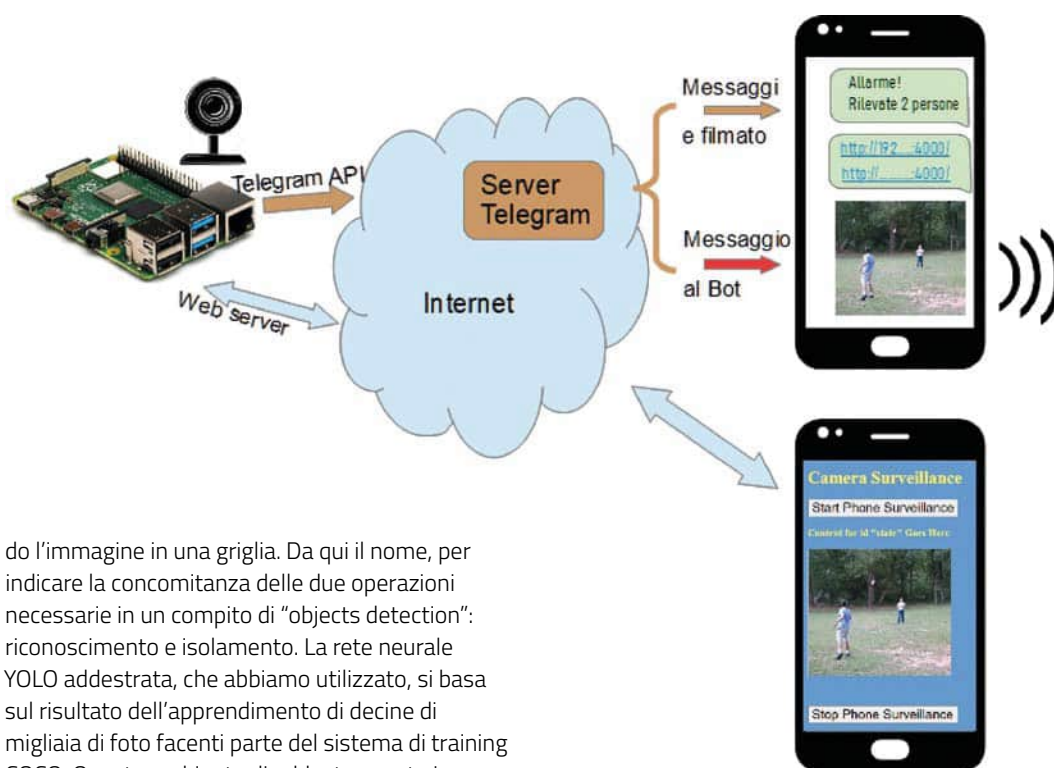
ALLARME ANTI-INTRUSIONE CON AI

dell'ing. DANIELE DENARO

Sistema di sicurezza basato sul riconoscimento d'immagine implementato eseguendo su una Raspberry Pi degli algoritmi di intelligenza artificiale.

Di recente vi abbiamo spiegato come realizzare un sistema di allarme basato sulla visione artificiale, implementata da una telecamera e da un sistema elettronico che discrimina la figura umana da altre possibili sagome. In questo modo il sistema può essere immune da falsi allarmi dovuti ad animali o a movimenti di oggetti di altra natura, come le fronde mosse dal vento, quando la telecamera è rivolta all'esterno. Infatti i sistemi di sorveglianza con telecamera, in genere hanno un funzionamento basilare che rileva la modifica di pixel in un'area più o meno significativa del frame. Invece per discriminare, come fa il nostro sistema, ci siamo rivolti alle tecniche di machine learning ed in particolare alla struttura deep learning "YOLO (You Only Look Once)". Questo algoritmo non solo utilizza strati convolutivi per identificare varie sagome, ma le localizza anche, suddividen-

Fig. 1
Allarme
anti-intrusione
intelligente



do l'immagine in una griglia. Da qui il nome, per indicare la concomitanza delle due operazioni necessarie in un compito di "objects detection": riconoscimento e isolamento. La rete neurale YOLO addestrata, che abbiamo utilizzato, si basa sul risultato dell'apprendimento di decine di migliaia di foto facenti parte del sistema di training COCO. Questo ambiente di addestramento insegna alla rete a riconoscere e localizzare 80 categorie di oggetti, fra cui le persone (Person), descritte nell'apposito riquadro.

Poiché questi sistemi deep-learning sono molto pesanti in termini di calcolo, utilizzeremo la recente Raspberry Pi 4. Nel nostro caso serve una frequenza di rilevamento di almeno 1 f/s, perché la mobilità di un intruso umano è in questo ordine di grandezza. Il processore ARM quadri-core della Raspberry Pi che è praticamente equivalente alla scheda Jetson Nano utilizzata nel precedente progetto, ma gli manca il motore a core multipli CUDA, particolarmente indicato per progetti con reti neurali deep learning. La memoria non è critica: noi abbiamo usato 8GB, ma il sistema funzionerebbe anche con 4GB, perché il collo di bottiglia è la capacità di calcolo.

Utilizzando la stessa struttura software implementata su Jetson Nano, su Raspberry Pi abbiamo bisogno di installare OpenCv; purtroppo va ricompilato a partire dal sorgente (che potete scaricare dal sito "opencv.org"). Questa operazione è abbastanza pesante e lenta da eseguire (richiede qualche ora) ma sul nostro sito potete trovare la versione già predisposta e compilata allo scopo. Prima di procedere apriamo quindi una parentesi sull'installazione di OpenCv in ambiente Raspberry Pi, dandovi qui di seguito le indicazioni che servono a implementarla.

OPENCV SU RASPBERRY PI

OpenCV è un noto software open-source per la elaborazione di immagini, che abbiamo già descritto in passato e che è praticamente fondamentale per questo tipo di applicazioni. Siccome le ultime versioni permettono anche di caricare ed eseguire diversi tipi di reti deep learning, il ciclo di utilizzo diventa completo: acquisizione, pre-trattamento, caricamento come input della rete, esecuzione e post elaborazione.

Purtroppo avere OpenCv su Raspberry Pi non è semplice, a meno di limitarsi alle funzioni base e non ottimizzate. Occorre infatti partire dai sorgenti e compilare il tutto. Fortunatamente esiste una procedura automatizzata per la compilazione con "cmake" ma per contro il numero di parametri da gestire è veramente esagerato: ci sono dei valori predefiniti, ma non sempre vanno bene per una buona ottimizzazione. Quindi è necessario esaminarli uno ad uno, sfruttando una applicazione cmake grafica come "cmake-gui". Per questo motivo si è cercato di preparare una versione precompilata con i parametri opportuni da installare su Raspberry Pi.

OpenCv su Raspberry Pi può essere scaricato dal repository https://github.com/open-electronics/Artificial_Intelligence/tree/master/Deep_Learning/

TIPOLOGIE DI OGGETTI RICONOSCIUTE

La rete YOLOv4-tiny può identificare una varietà di elementi nell'immagine, che sono quelli elencati qui di seguito (i nomi sono tutti in inglese).

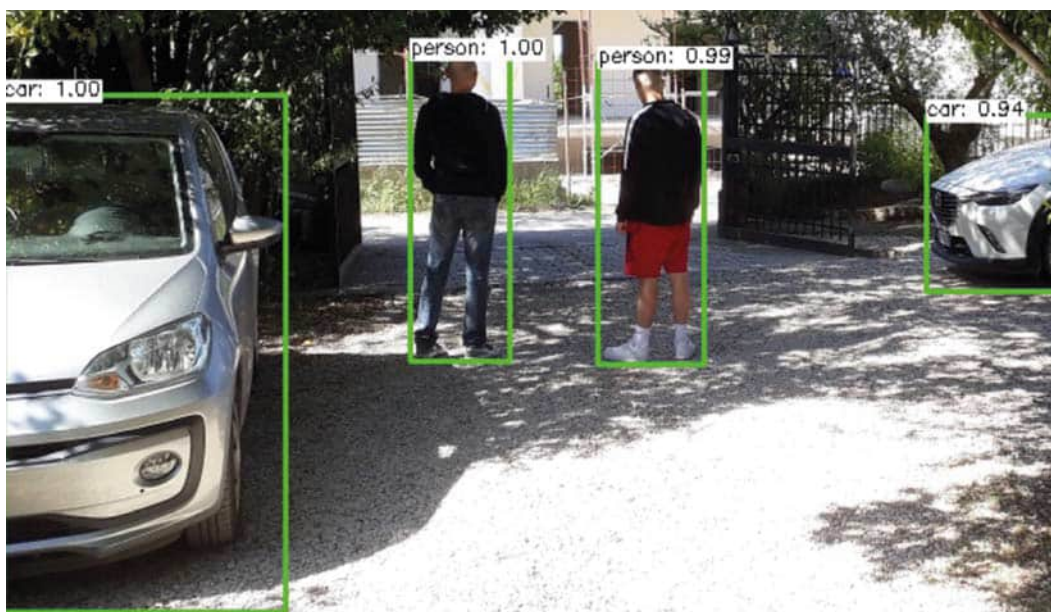
person, bicycle, car, motorbike, aeroplane, bus, train, truck, boat, traffic, light, fire, hydrant, stop, sign, parking, meter, bench, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard, sports, ball, kite, baseball, bat, baseball, glove, skateboard, surfboard, tennis, racket, bottle, wine, glass, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, hot, dog, pizza, donut, cake, chair, sofa, pottedplant, bed, diningtable, toilet, tvmonitor, laptop, mouse, remote, keyboard, cell, phone, microwave, oven, toaster, sink, refrigerator, book, clock, vase, scissors, teddy, bear, hair, drier, toothbrush.



YoloPythonRasp nella sottodirectory "OpenCV4Raspberry" che contiene un link per scaricare il software. Il software non è contenuto in github a causa del suo peso (450MB).

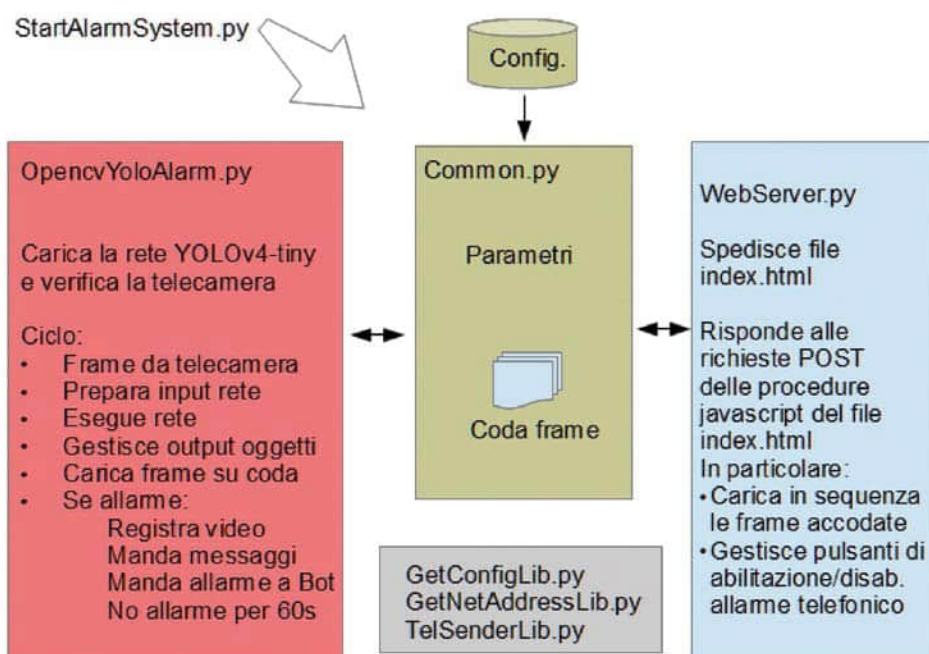
Scaricate quindi la cartella complessiva "OpenCv-4.5.2" nella directory /home di Raspberry Pi; tale cartella comprende due ulteriori cartelle

corrispondenti a "opencv-4.5.2" e "opencv-contrib-master". La prima contiene il sistema OpenCv vero e proprio, mentre la seconda contiene diversi moduli aggiuntivi come per esempio quello relativo alle reti deep-learning (dnn). La cartella di OpenCv contiene non solo i sorgenti ma anche i file compilati e la libreria completa.



Esempio di rilevamento.

Fig. 2
Schema a blocchi
del software.



Se il sistema operativo è quello recente, i file compilati dovrebbero andar bene.

Allora posizionatevi nella sotto-cartella "OpenCv-4.5.2/opencv-4.5.2/build" e lanciate il comando "sudo make install".

Il comando dovrebbe fare solo l'installazione delle librerie in "/usr/local". In caso di problemi, bisognerà eseguire la procedura completa. Installate (se non già presente) l'applicazione "cmake-gui" con il comando "sudo apt-get install cmake-gui".

E' sempre opportuno prima delle installazioni Linux lanciare i comandi "sudo apt-get update" e poi "sudo apt-get upgrade", per mantenere aggiornato il sistema. A questo punto posizionatevi in "OpenCv-4.5.2/opencv-4.5.2" e lanciate l'applicazione "cmake-gui".

Cmake-gui è un'utile interfaccia grafica che permette di attivare/disattivare i numerosi parametri di compilazione di Opencv attingendo le informazioni dal file "CmakeLists.txt".

Questo file è stato già modificato con i parametri utili per Raspberry Pi.

Nell'interfaccia grafica cliccate sul pulsante <Configure> e leggete l'output che produce; se non si verificano particolari problemi attivate anche il pulsante <Generate>. Così facendo, nella directory "build" sarà stato aggiornato il file "makefile", quindi, da questa directory, potrete impartire di nuovo il comando:

```
sudo make install
```

In realtà è meglio impartire il comando completo:

```
nohup sudo make install -j4 &
```

In questo modo l'installazione avviene in background e l'output del processo di compilazione-installazione apparirà nel file "nohup.out".

Il parametro "-j4" permette alla compilazione di utilizzare tutti e 4 i core di Raspberry Pi.

Comunque il processo di compilazione può durare un paio di ore.

Con 8GB di memoria non ci sono particolari problemi, ma con dimensioni minori è meglio ampliare il file di SWAP (memoria virtuale). Per far questo bisogna aprire con un editor (ma in modalità super user) il file /etc/dphys-swapfile e modificare la voce CONF_SWAPSIZE da 100 (MB) a 2000. Salvo riportare lo swap file alle dimensioni di default alla fine dell'installazione.

Alcuni parametri utilizzati per la compilazione di OpenCv sono: OPENMP (abilita il multiprocessing), NEON e VFPV3 che sono acceleratori per le architetture ARMv7.

Per compilare OpenCv con la gestione dei filmati "mp4" cioè con il parametro FFMPEG, dovrete avere già installato le seguenti librerie: **libavcode**, **libavformat**, **libsvscale**, **libavutil** e **libdc1394**.

Potete installarle con il solito "apt-get install" o più semplicemente utilizzando l'applicazione "synaptic" che permette una facile e potente ricerca ed installazione del software Linux.

LE PRESTAZIONI CON RASPBERRY PI

Bene, chiarito come installare e configurare Opencv sulla Raspberry Pi 4 riprendiamo il discorso e spieghiamo cosa abbiamo riscontrato nell'utilizzo: la velocità di esecuzione della rete YOLO si è dimostrata decisamente inferiore. Se con Jetson abbiamo avuto performance maggiori di 1f/s utilizzando Yolo versione 4, e maggiori di 15f/s utilizzando la sua versione ridotta (tiny), con Raspberry Pi abbiamo potuto utilizzare solo la versione tiny, con un frame rate di circa 1,6 f/s. Con la versione tiny, un po' meno precisa, abbiamo comunque la possibilità di utilizzare Raspberry Pi come allarme anti-intrusione perché il frame rate è sufficiente per i nostri scopi.

Il funzionamento è identico a quello su Jetson Nano, ma i consumi sono inferiori.

In particolare su Raspberry Pi 4 con 8GB di memoria si sono misurati:

- 0,5/0,7 A in attività normali;
- 1 A con l'applicazione di rilevamento ed allarme attiva (YOLO4-tiny: 1,6f/s).

Quindi il sistema attivo consumerebbe circa 5W.

Nel nostro sito potete trovare una cartella specifica per il sistema di allarme su Raspberry Pi, anche se è praticamente uguale a quello per Jetson Nano.

https://github.com/open-electronics/Artificial_Intelligence/tree/master/Deep_Learning/YoloPythonRasp E poi nella sotto-cartella "OpencvYoloAlarm".

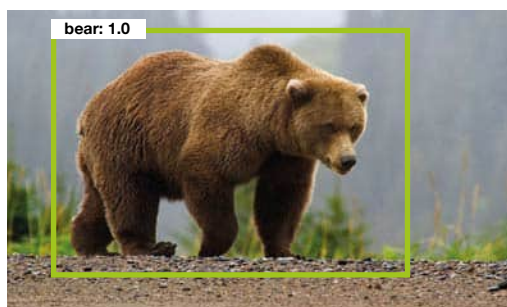
Per chi non avesse letto l'articolo sul precedente numero della rivista, il file di configurazione "alarmssystem.cfg" va aggiornato con i dati della registrazione su Telegram. Tra breve spiegheremo come fare.

Sempre per chi non avesse letto l'articolo precedente, riassumiamo brevemente il funzionamento del sistema.

Su Raspberry Pi, il sistema esamina ogni frame ricevuto dalla telecamera (che può essere ovviamente una Pi-camera) e la processa tramite la rete YOLOv4-tiny, producendo un output dove vengono identificati, se presenti, oggetti classificati in 80 tipologie che riportiamo in un apposito riquadro in queste pagine.

Se tra questi ci sono uno o più "person", attiva la modalità allarme e si astiene per 60 secondi successivi, per evitare allarmi multipli. (L'intervallo può essere cambiato sul file di configurazione). In collaborazione con il sistema chat di Telegram, l'allarme consiste in:

- registrazione di 10s di filmato incriminato;
- invio dell'indirizzo locale e di quello pubblico;



- invio di un messaggio con il rilevamento del numero di oggetti rilevati;
- invio del filmato;
- invio di allarme sul bot attivato nel proprio telefonino (che attiva suoneria).

L'invio degli indirizzi Internet permette di collegarsi immediatamente tramite il web server inserito nel sistema, nel caso si volessero approfondire le indicazioni fornite dal filmato. L'indirizzo pubblico è raggiungibile dappertutto a patto che sul router casalingo sia stato fatto il "Port Forwarding" di Raspberry Pi. Il web server è minimale e utilizza le librerie standard di Python. La porta a cui fare riferimento è la porta 4000.

Nel file (testuale) di configurazione "alarmssystem.cfg" oltre ai codici per il collegamento con Telegram e al periodo di sospensione dei segnali di allarme, è possibile cambiare la classe dell'oggetto e la soglia di confidenza che attiva l'allarme. Infatti il sistema può essere utilizzato anche per altri scopi, come per esempio le trappole fotografiche (**Fig. 3**).

Poiché il sistema è implementato in Python, che è un linguaggio interpretato, è facilmente modificabile nella routine "sendAlarm()" per eliminare l'invio di messaggi e trasformare la routine in una semplice attivazione di un relé per accendere, per esempio, l'illuminazione di un giardino in seguito



Fig. 3
Trappola fotografica.

Fig. 4
Accensione luci in caso di presenza umana.

alla sola presenza umana (**Fig. 4**).

Nella cartella del sito sono presenti altri due programmi:

- OpenCvYoloCamera, che analizza e visualizza in modo continuo ciò che la telecamera vede; non ha la stessa fluidità della versione per Jetson Nano che può contare sui core CUDA;
- OpenCvYoloPhoto, che permette di scegliere delle foto da analizzare producendo una versione con gli oggetti inquadrati ed una lista degli oggetti rilevati.

API TELEGRAM PER PYTHON

Per avvertire dell'avvenuta intrusione il sistema si basa sull'onnipresente Internet. I sistemi di sorveglianza commerciali mettono a disposizione un loro server che instrada gli allarmi verso applicazioni smartphone proprietarie. Noi, per questo sistema, sfrutteremo i sistemi di chat già esistenti. In particolare useremo Telegram. La scelta di Telegram, al posto di WhatsApp, è dovuta sostanzialmente a due fattori: la presenza di semplici API Python per l'utilizzo di Telegram e la possibilità di creare dei BOT. Anche se esistono API per WhatsApp, rimane il problema di attivare una suoneria di allarme. Infatti sia WhatsApp che Telegram non attivano notifiche quando viene spedito un messaggio al proprio account: spedire un messaggio a se stessi, anche se da un client decentrato, è necessario per evitare una ulteriore SIM. Ma Telegram permette la creazione di BOT, che sono dei client virtuali e possono girare ovunque, anche sul proprio telefonino. Il BOT sul telefonino ci servirà per lanciare una suoneria mp3 a nostra scelta.

Il sistema di allarme utilizza una semplice libreria Python per Telegram che si chiama "Telethon" <https://pypi.org/project/Telethon/>.

La libreria va installata su Raspberry Pi con il solito comando:

```
pip3 install telethon
```

A questo punto bisogna creare un account Telegram (se non lo si ha già) dove arriveranno i messaggi. Ma questo non basta per essere abilitati ad utilizzare le API e spedire messaggi da una applicazione.

Occorre farsi dare dei codici e in particolare:

- Application Id (codice numerico);
- Chiave crittografica hash (codice alfanumerico).

Questi due codici vanno richiesti al sito <https://my.telegram.org/> presentandosi con il proprio

account (numero di telefono). I codici andranno inseriti nel file di configurazione "alarmssystem.cfg" nelle voci predisposte. A questo punto il sistema di allarme può inviare messaggi al nostro account Telegram. Per poter attivare un forte segnale di allarme sul nostro telefonino dobbiamo però costruire ed installare un BOT. Anche in questo caso servono dei codici.

Si possono ottenere dall'utenza Telegram (sul telefonino o sul client WEB). Su ogni utenza Telegram è presente un riferimento "BotFather" nella lista delle chat. Questo riferimento permette di attivare un nuovo BOT. Poiché anch'esso è un BOT (il padre di tutti i BOT), si possono usare i comandi previsti nel formato "/comando" ovvero nel nostro caso : "/newbot".

A questo punto BotFather vi chiederà di inserire un nome di riferimento per il vostro BOT e soprattutto il Nick-name (indirizzo) il quale può essere un qualunque nome che deve però finire con ...bot (ad esempio miobot, mioprimo_bot ecc). Ad esso BotFather risponderà fornendovi una Chiave crittografica hash relativa al particolare BOT creato.

Anche il Nick-name va inserito nel file di configurazione (nel formato @nomebot"), perché sarà l'indirizzo del BOT a cui invieremo l'allarme, mentre il codice hash va inserito nello script che implementerà il BOT sullo smartphone.

La libreria "telethon" prevede di creare un file di sessione al primo collegamento, per evitare di inserire ogni volta il nostro numero di telefono. Per creare questa sessione al di fuori dell'applicazione di allarme è stato inserito uno script che va lanciato una sola volta. Lo script "TelegramSignIn.py" legge i codici dal file di configurazione, chiede, per verifica, il numero di telefono e crea la sessione su Raspberry Pi.

IL BOT SUL TELEFONINO ANDROID

Una volta che sono stati definiti il Nick-name e la chiave hash del BOT, bisogna implementare il BOT stesso sullo smartphone. Nella sotto-cartella "PhoneScript" del sistema di allarme, si trova lo script "BotAlarm.py". Per lanciare lo script dobbiamo però caricare Python per Android. Questo sarà possibile solo installando da "GooglePlay" l'applicazione "Termux". Termux permette di aprire una console Linux su Android (che comunque si basa su Linux). Dalla console Linux (che comunque non necessita dei privilegi di "root") installiamo prima Python con il comando "pkg install python", poi telethon con il comando "pip install telethon" ed infine un esecutore di file mp3 come play-audio



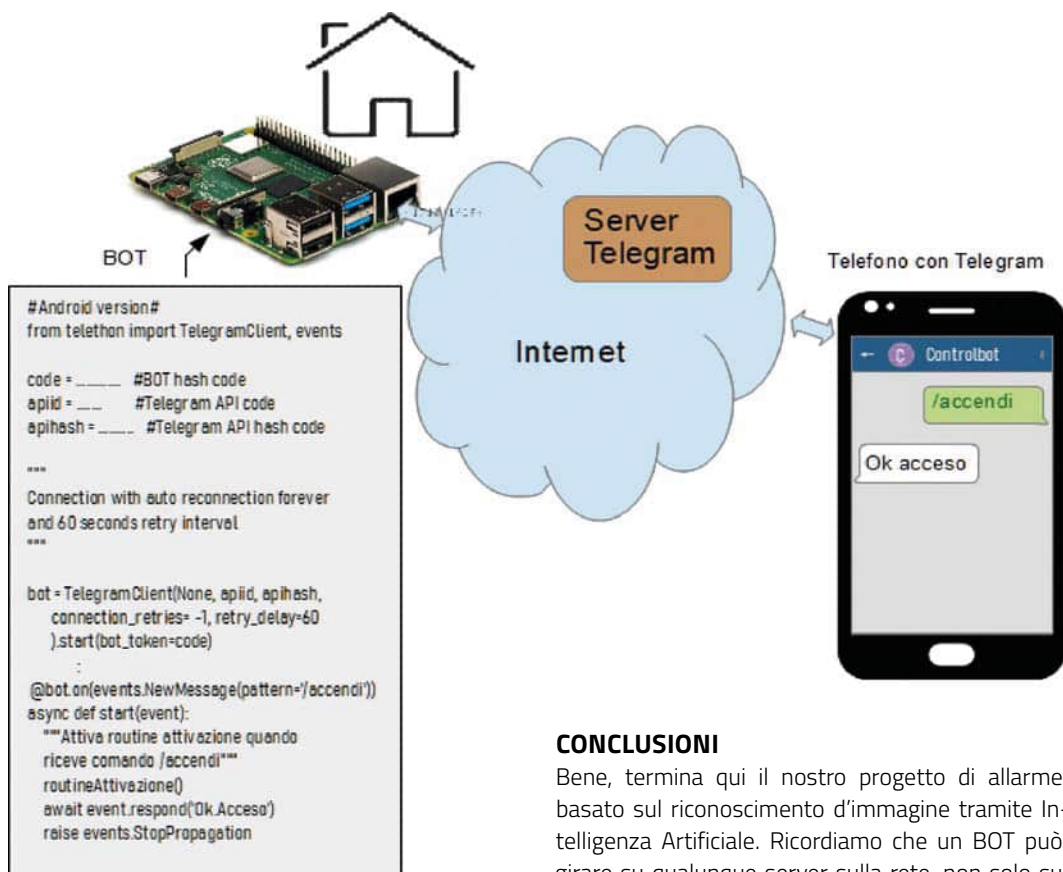


Fig. 5
Esempio di
possibile
comando
remoto
tramite
Telegram.

con il comando "pkg install play-audio". Lo script va modificato inserendoci i codici opportuni forniti da Telegram. Una volta lanciato lo script dalla console Termux con il comando "python BotAlarm.py", questo rimarrà in ascolto del comando che gli farà azionare la suoneria (nella cartella è compreso il file "BotAlarm.mp3" che potete ovviamente cambiare con uno di vostro gradimento). Non si è voluto implementare una applicazione Android, che avrebbe sicuramente semplificato l'attivazione del BOT, per rendere visibile e modificabile ogni linea di codice presente sul vostro smartphone nello spirito delle applicazioni hobbistiche. Applicazioni Android che fanno fare cose al vostro telefono, magari da remoto, non sono consigliabili, ed infatti non le troverete su Google Play, perché potrebbero nascondere attività di spionaggio nei vostri confronti. Ma non escludiamo che in futuro potremmo realizzare una procedura per creare questa applicazione da installare sul vostro telefono ma lasciando all'utente il compito di costruirla in base ai moduli ed alle procedure fornite, così da poterne essere confidente.

CONCLUSIONI

Bene, termina qui il nostro progetto di allarme basato sul riconoscimento d'immagine tramite Intelligenza Artificiale. Ricordiamo che un BOT può girare su qualunque server sulla rete, non solo su un dispositivo mobile come lo smartphone. Anzi, la maggior parte degli utilizzi dei BOT è quello di fornire, tramite chat, servizi di varia natura quando interpellati. Per questo possono risiedere anche sul server Telegram. Nulla vieta, quindi di utilizzare un nostro computer come Raspberry Pi come controllore remoto casalingo tramite telefonino: per esempio portando lo script dell'allarme su Raspberry Pi e codificando diversamente la risposta ad un comando, potremmo accendere il riscaldamento o altro.

In **Fig. 5** potete vedere uno schema generale ed un estratto di un possibile BOT scritto in Python. ■



Cosa occorre?

Raspberry Pi 4 Tipo B con 4GB di memoria (cod. RPI4-4GB) è disponibile da Futura Elettronica a Euro 69,00, la stessa versione con 8GB di memoria (cod. RPI4-8GB) è in vendita a Euro 99,00, la fotocamera da 5 megapixel (cod. YG016) è disponibile a Euro 9,50.

Per Raspberry Pi 4 Tipo B sono disponibili anche diversi starter kit già pronti all'utilizzo disponibili alla pagina bit.ly/RPIStarterKit

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

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

UN TEAM DI PROFESSIONISTI PER IL TUO PROGETTO



oltre **1500** progetti sviluppati
26 anni di attività
più di **15** progettisti



L'area R&D di Futura Group studia, progetta e realizza prototipi per differenti settori utilizzando tecnologie innovative. Nel team di Futura Group fa parte il personale interno e un numero elevato di collaboratori esterni, ciascuno specializzato nel proprio settore (Disegno CAD – Raspberry Pi – Arduino – Audio – Sviluppo firmware – Linux – ecc). La disponibilità di oltre 5000 prodotti elettronici e componenti consente di ridurre i tempi di test e sviluppo e di realizzare facilmente il primo prototipo. La decennale esperienza nel settore e la necessità di ottimizzare i tempi e i costi, ha portato Futura Group a selezionare una serie di terzisti per la produzione di piccola o grande serie.

I SERVIZI CHE TI OFFRIAMO



Progetto Completo

Partendo dalla tua idea progettiamo per te la soluzione.



Sviluppo PCB

Ottimizziamo il tuo prototipo disegnando un PCB compatto secondo le tue specifiche.



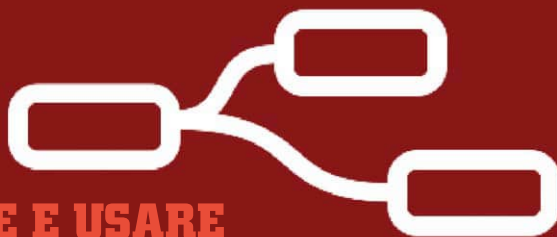
Scrittura Codice

Scriviamo il codice del core per dare vita alla tua applicazione.



Produzione Progetto

Portiamo il prototipo alla fase di produzione seguendo per te tutte le fasi.



CONOSCERE E USARE

di FRANCESCO FICILI

1

Node-RED

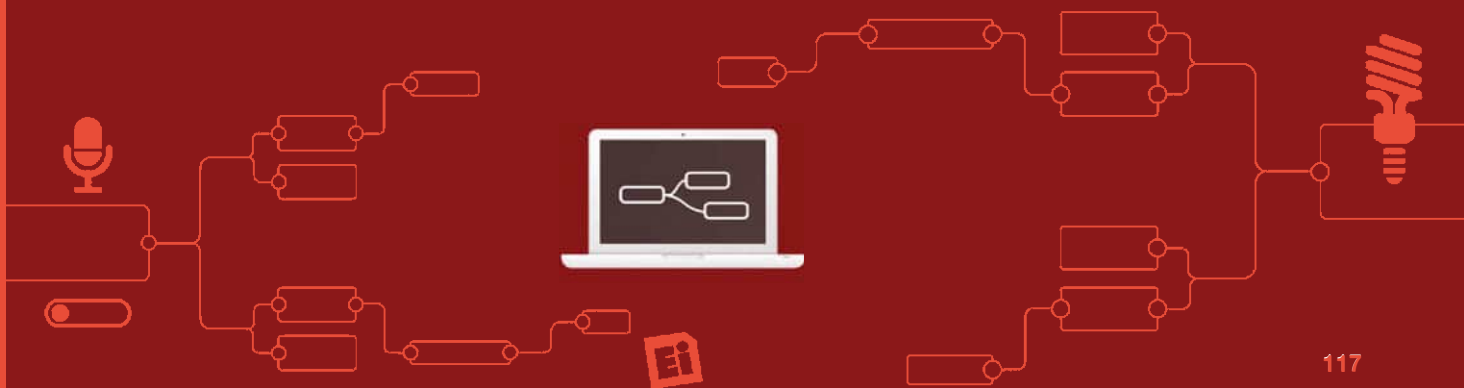
Iniziamo a scoprire un tool di flow-based programming, orientato all'IoT ed alla connettività, originariamente sviluppato dall' IBM Emerging Technology Services team e adesso parte della JS Foundation. Spiegheremo cos'è Node-RED ed inizieremo a prendere confidenza con l'ambiente e con l'editor grafico, mostrando poi alcuni esempi di utilizzo su Raspberry PI.

Prima Puntata.

Negli ultimi anni si è assistito ad una sempre maggiore diffusione di applicazioni IoT, tanto che ormai la connettività è una caratteristica di base della maggior parte degli oggetti che utilizziamo ogni giorno. Basti pensare ad un comune elettrodomestico moderno, come una lavatrice o un frigorifero, per renderci conto che ormai il fatto che l'oggetto sia connesso è una caratteristica quasi imprescindibile in ogni genere di applicazioni, per non parlare di oggetti più specializzati, come ad esempio gli assistenti vocali di Amazon e Google, che fanno della connettività e della loro vocazione IoT la loro caratteristica principale. Anche nel mondo dei maker e degli hobbisti si percepisce sempre più la necessità di poter connettere i propri progetti con la Rete, per eseguire le più svariate funzionalità; pensiamo ad esempio ad un impianto di irrigazione che desideriamo controllare dal nostro smartphone o ad una stazione meteo

che ci fornisca i dati acquisiti dai suoi sensori direttamente su un PC o via mail, o, ancora meglio, tramite Telegram.

Utilizzando piattaforme ed ambienti di sviluppo tradizionali è sicuramente possibile ottenere risultati di questo tipo, ma è necessario scrivere parecchie righe di codice, utilizzando librerie di una certa complessità che spesso pregiudicano la manutenibilità e la scalabilità delle nostre applicazioni. Una soluzione alternativa al problema, che semplifica moltissimo l'approccio e consente di avere delle applicazioni sempre facilmente leggibili e scalabili, è l'utilizzo di un ambiente grafico specializzato nella realizzazione di applicazioni IoT ed orientate alla connettività. Dato che questa necessità è fortemente sentita, un gruppo di programmatori della IBM ha sviluppato un ambiente di programmazione che segue questo approccio, dando vita a quello che oggi è conosciuto come Node-RED.



COS'È NODE-RED

Node-RED è un tool di programmazione grafica sviluppato dall'Emerging Technology Services team di IBM. Il progetto nasce all'inizio del 2013, da Nick O'Leary e Dave Conway-Jones, come un proof-of-concepts per la visualizzazione ed il mapping tra topics MQTT, per diventare velocemente un tool più generale che consente lo sviluppo di applicazioni di vario tipo. Il progetto è open-source praticamente dalla sua concezione, e quest'orientamento open è culminato nel divenire uno dei 32 progetti della JS foundation, nell'ottobre 2016.

Node-RED consiste sostanzialmente in una runtime di *Node.js*, che implementa un server al quale si può far puntare un qualsiasi browser per accedere all'editor grafico. All'interno dell'editor visualizzato dal browser è possibile creare i programmi sotto forma di insieme di nodi (flussi o flows) interconnessi tra di loro.

I nodi vengono inseriti nei flussi prelevandoli da un'apposita palette presente nell'editor. La palette dei nodi può facilmente essere espansa tramite altri nodi creati dall'utente stesso o dalla community.

Una volta che il flusso è completo è possibile eseguirne il deploy e quindi mandarlo in esecuzione, tramite un apposito pulsante.

Essendo basato su Node.js, Node-RED può virtualmente

essere eseguito su qualsiasi sistema in grado di supportare Node.js, compresi:

- PC Linux o Windows;
- sistemi embedded come Raspberry Pi o BeagleBoard;
- sistemi Android;
- server.

Nel resto di questo corso utilizzeremo Node-RED su Raspberry Pi, ma i concetti esposti restano validi per l'utilizzo di Node-RED anche su altre piattaforme (sono riassunte nella Fig. 2); ciò che può cambiare sono principalmente alcune caratteristiche relative all'installazione, per le quali rimandiamo i lettori all'apposita sezione sul sito web ufficiale: <https://nodered.org/docs/getting-started/>.

FLOW-BASED PROGRAMMING

Il concetto principale su cui si basa Node-RED è quello della cosiddetta "Flow-Based Programming" o FBP. Tale paradigma di programmazione, enunciato per la prima volta da J. Paul Morrison nel 1970, descrive le applicazioni come una network di processi "black box", ossia di cui non si conoscono i dettagli implementativi, ma di cui sono noti i dati in ingresso, il tipo di elaborazione eseguita su di essi e i dati in uscita. I processi black box sono anche noti come nodi (questo è proprio il termine utilizzato da Node-RED) e tali nodi



Fig. 1 - Logo di Node-RED

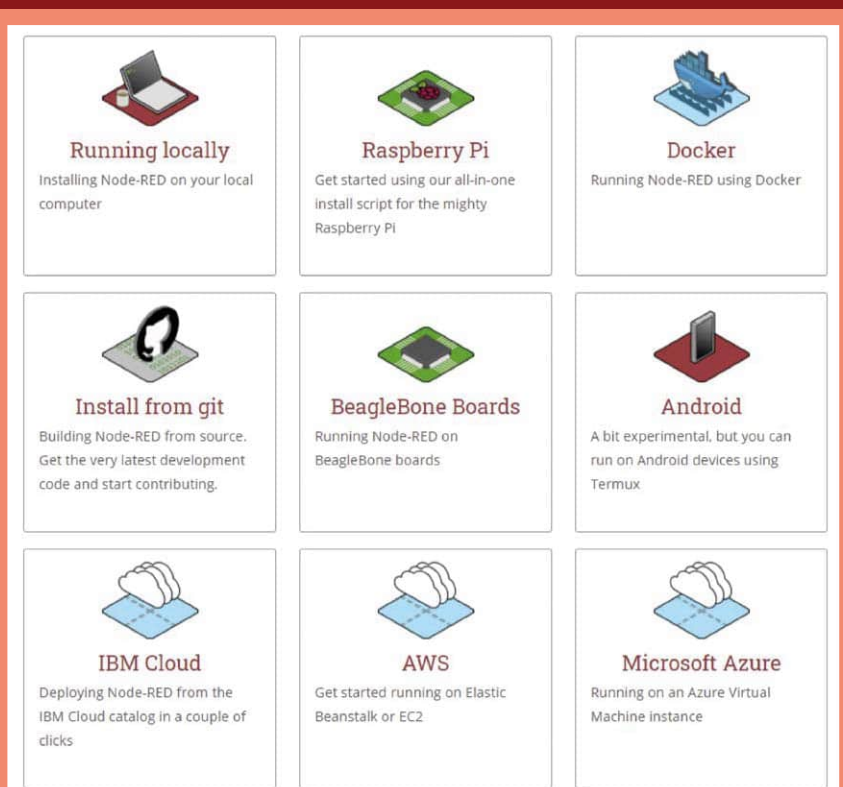


Fig. 2 - Le varie piattaforme che supportano l'esecuzione di Node-RED.

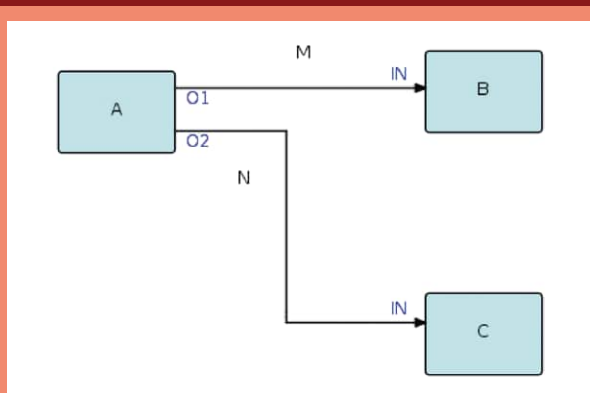


Fig. 3 - Un semplice diagramma di Flow-Based Programming.

si scambiano dati attraverso messaggi instradati tramite connessioni predefinite (network). Ne consegue che la FBP è naturalmente "component oriented", ossia è un paradigma che enfatizza la modularità nello sviluppo software. La FBP vede un'applicazione non come un processo sequenziale, che inizia ad un certo istante di tempo e fa una cosa alla volta finché non completa la sua esecuzione, ma piuttosto come un insieme di processi asincroni che comunicano tra di loro per mezzo di uno stream di dati strutturati, chiamati "information packets" (IPs). Il focus è quindi sui dati applicativi e sulle trasformazioni

applicate su di essi al fine di produrre l'output desiderato. La network è definita esternamente ai processi e viene interpretata a sua volta come una parte di software, normalmente chiamata "scheduler".

In FBP i processi comunicano per mezzo di connessioni e tali connessioni sono collegate ai processi per mezzo di porte (in ingresso o in uscita); in un certo lasso temporale, un IP può essere manipolato solo da un singolo processo, oppure può essere in transito tra due processi.

Data la sua natura, la definizione di una network in FBP è normalmente un diagramma, ma può anche essere convertita in una lista di connessioni e nodi, se viene descritta con un linguaggio di basso livello o una particolare notazione (vedremo che questo avviene anche in Node-RED).

In Fig. 3 è riportato un semplice diagramma FBP.

A, B e C sono processi, mentre O1, O2 e IN sono porte. M ed N invece sono connessioni che connettono le porte dei processi tra di loro. Le connessioni hanno una capacità fissa in termini di numero di IPs che possono essere immagazzinati in un certo istante di tempo.

Node-RED implementa tutti i concetti base della FBP, quindi un programma o, per usare la terminologia Node-RED, un flow, è descritto come una serie di nodi interconnessi tra di loro. Ogni nodo implementa una specifica funzione, riceve un set di dati da una porta in ingresso, li manipola e li passa

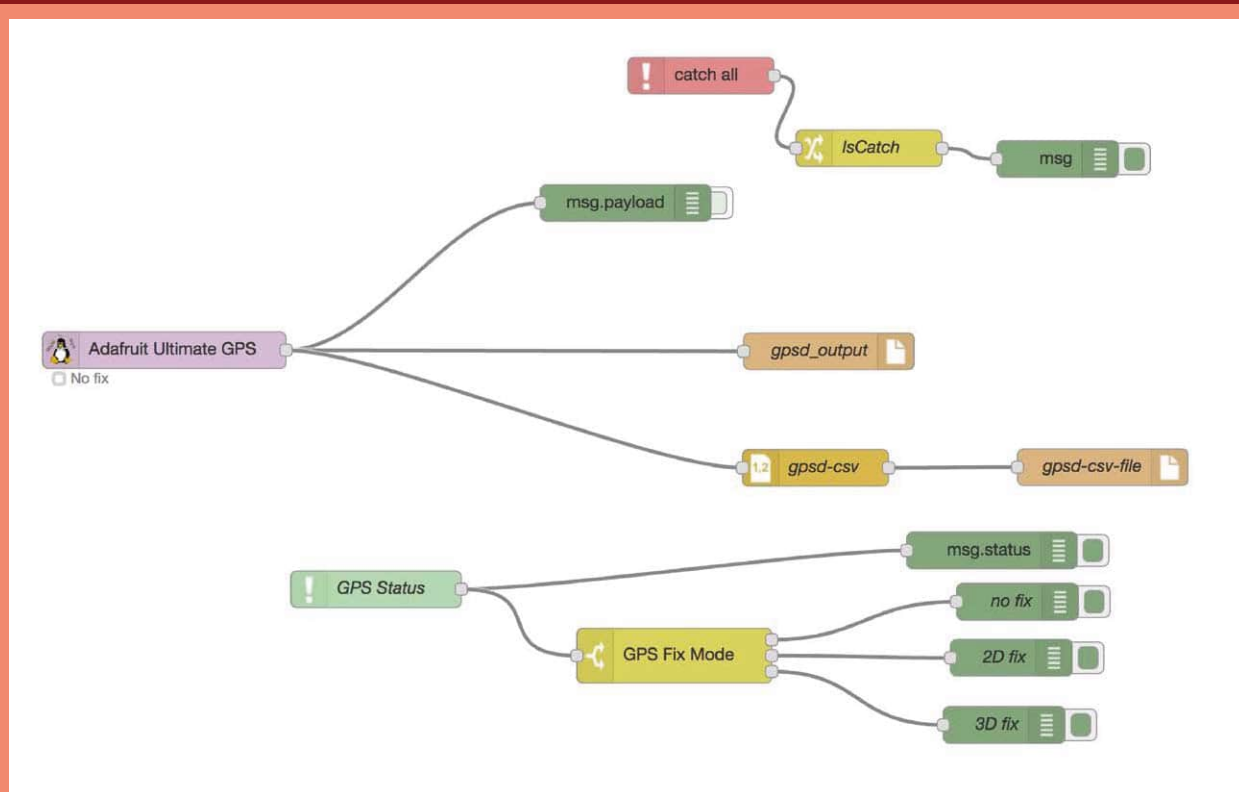


Fig. 4 - Esempio di Flow in Node-RED.

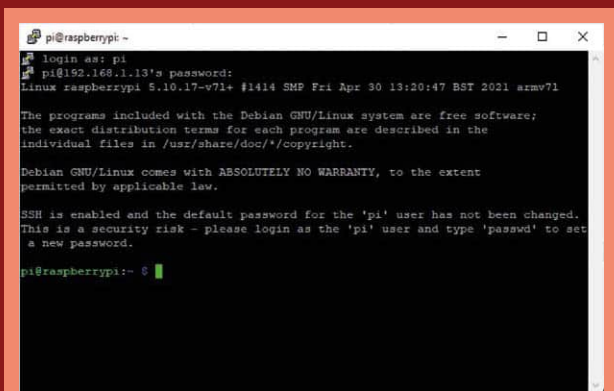


Fig. 5 - Schermata della console di Raspberry Pi Os visualizzata in SSH tramite PuTTY.

alla porta/e in uscita. La network è responsabile del flusso di dati attraverso i vari nodi. In Fig. 4 è riportato un esempio di flow in Node-RED.

Prima di addentrarci sui dettagli della programmazione di un flow vediamo però come installare ed avviare l'ambiente.

INSTALLARE ED AVVIARE NODE-RED

Come abbiamo già accennato, Node-RED può girare su una moltitudine di piattaforme: di fatto, su ogni piattaforma capace di far girare Node.js. Dal punto di vista dell'utilizzo non c'è una grossa differenza, ma ovviamente la procedura di installazione può differire da caso a caso.

Oltre a questo in alcuni casi occorre preparare opportunamente la piattaforma. In questa sezione forniremo una serie di istruzioni complete per l'installazione su Raspberry Pi, ma si tenga conto che non si tratta di istruzioni valide in generale, quindi consultate la relativa sezione del sito web di Node-RED per maggiori informazioni.

Per installare ed eseguire Node-RED su Raspberry Pi occorre innanzitutto munirsi di una Raspberry. Non ci sono particolari limitazioni rispetto al modello da utilizzare, ma si consiglia di non usare un modello troppo antiquato.

Per le nostre prove abbiamo utilizzato una Raspberry Pi 4 Model B con 4 GB di RAM.

Per prima cosa occorre preparare una scheda SD con un'immagine di Raspberry OS al suo interno, usando Raspberry Pi Imager. Se si utilizza un'immagine "full with recommended SW", Node-RED risulterà già preinstallato, ma sconsigliamo di utilizzare questa opzione sia per i tempi di creazione dell'immagine che per il fatto che la versione preinstallata di Node-RED potrebbe essere non aggiornata all'ultima versione.

In alternativa è possibile creare un'immagine di tipo Lite ed installare Node-RED successivamente.

Creiamo quindi un'immagine di tipo Lite e predisponiamo la Raspberry al collegamento con la nostra rete domestica in modalità headless, abilitando al contempo la connessione SSH. Per l'accesso headless rimandiamo alla guida reperibi-

le su <https://www.raspberrypi.org/documentation/configuration/wireless/headless.md>

presente sul sito web della raspberry Pi foundation. Per abilitare la connessione SSH inserire un file privo di estensione e con nome "SSH" all'interno della partizione di boot della SD-card. Una volta eseguite queste operazioni sarà possibile collegarsi alla Raspberry Pi tramite un client SSH, come ad esempio PuTTY (Fig. 5).

Per installare Node-RED e le sue dipendenze principali (ossia Node.js e npm) è possibile utilizzare il seguente script, che installa i vari componenti ed i nodi specifici per Raspberry Pi. Lo script può anche essere utilizzato per aggiornare un'installazione di Node-RED esistente.

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Più in dettaglio lo script esegue le seguenti operazioni:

- rimuove le versioni di Node-RED e Node.js preesistenti, se presenti;
- installa la versione di Node.js LTS corrente usando NodeSource; se Node.js è già presente viene eseguito un controllo di compatibilità per assicurarsi che Node sia almeno alla versione 12;
- installa l'ultima versione di Node-RED usando npm;
- se desiderato dall'utente installa una serie di nodi specifici per Raspberry Pi;
- setta Node-RED per funzionare come service e fornisce una serie di comandi specifici per utilizzare i servizi.

Il risultato dell'esecuzione è quello proposto nella schermata di Fig. 6. Si noti inoltre che Node-RED è stato anche inserito nei repository di Raspberry Pi Os, e compare nella lista dei software raccomandati dalla foundation, e può quindi essere installato anche tramite apt-get (apt-get

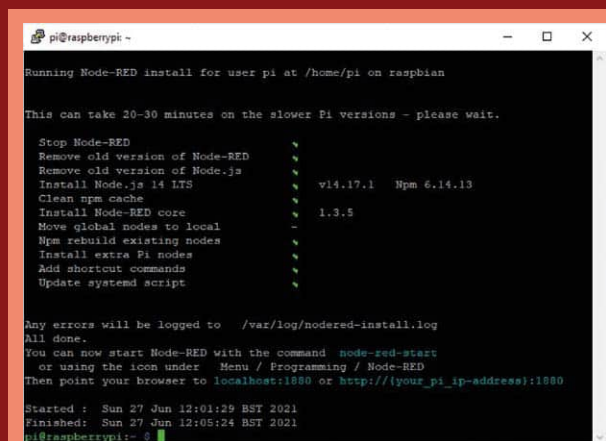


Fig. 6 - Schermata della console dopo l'esecuzione dello script

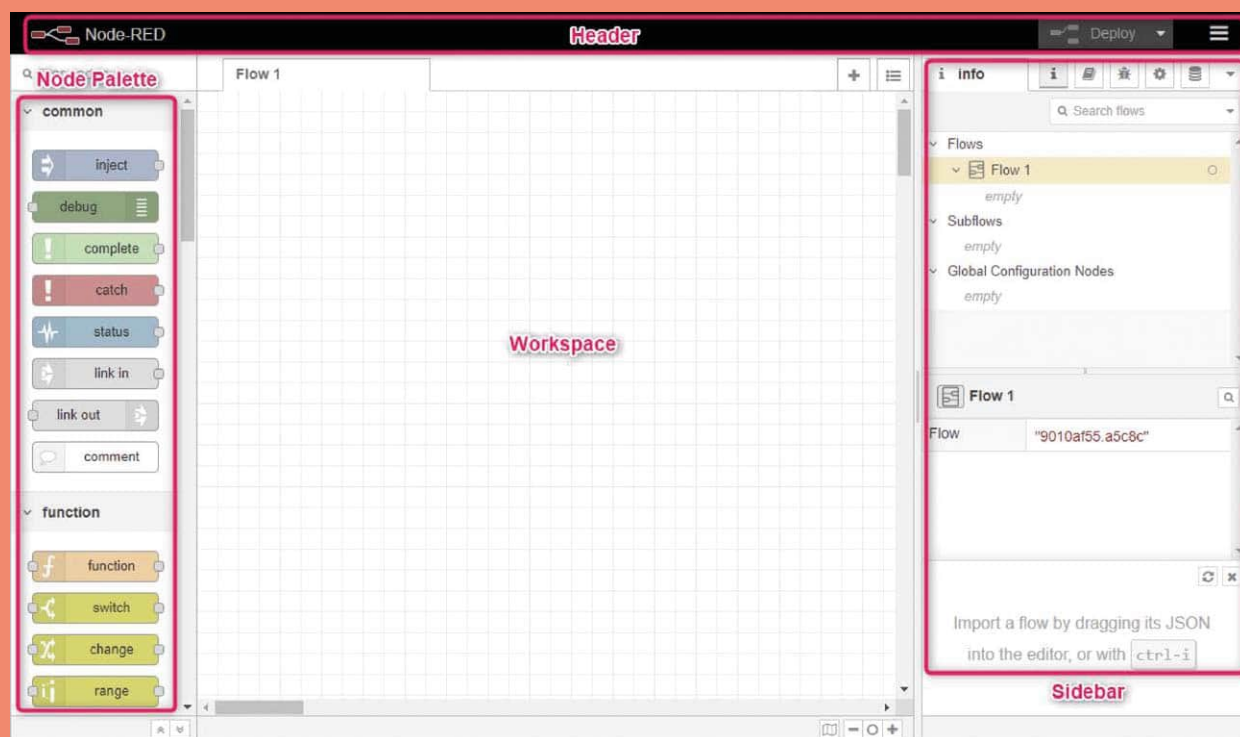


Fig. 7 - Editor di Node-RED.

install nodered). Tuttavia, con questa procedura non verrà installato npm (mentre è comunque incluso Node.js). La documentazione di Node-RED comunque suggerisce l'utilizzo dello script per installare ed aggiornare Node-RED su Raspberry Pi. Come spiegato, lo script esegue anche i settaggi necessari per permettere la gestione di Node-RED come servizio; ciò significa che è possibile far girare Node-RED in background ed avviarlo automaticamente al boot. Sono disponibili i seguenti comandi per la gestione del servizio:

- `node-red-start`, che avvia Node-RED e visualizza i log sulla console;
- `node-red-stop`, che arresta Node-RED;
- `node-red-restart`, che arresta e riavvia Node-RED;
- `node-red-log`, che visualizza i log del servizio.

È anche possibile fare in modo che il servizio venga avviato automaticamente al boot. Per farlo è sufficiente eseguire il seguente comando sulla console:

```
sudo systemctl enable nodered.service
```

Per disabilitare l'autostart al boot basta invece eseguire questo comando:

```
sudo systemctl disable nodered.service
```

NODE-RED EDITOR

Ora che Node-RED è correttamente installato sulla nostra Raspberry Pi, possiamo aprire l'editor; questo può essere aperto con un comune browser, puntando all'indirizzo nel quale il server sta girando. Quindi se vogliamo accedere "localmente", ossia dalla stessa Raspberry nella quale abbiamo installato Node-RED è sufficiente puntare alla porta 1880 del localhost: `http://localhost:1880`. Tuttavia nella maggior parte dei casi risulta più conveniente accedere a Node-RED da un altro PC, puntando all'indirizzo IP della Raspberry nella nostra network locale sulla porta 1880, ossia `http://<hostname>:1880`, dove `<hostname>` rappresenta appunto l'indirizzo IP della Raspberry, che può essere recuperato dal nostro access point casalingo. Una volta aperto il browser all'indirizzo corretto e sulla porta 1880, dovremmo trovare una schermata simile a quella in Fig. 7. L'editor è composto essenzialmente dai seguenti quattro componenti:

- **Header:** contiene il pulsante di deploy, il menu principale, e, se la user authentication è abilitata, lo user menu.
- **Workspace:** è l'area di lavoro sulla quale i flows possono essere creati, inserendo nodi e connessioni tra di essi.
- **Palette:** contiene i nodi che possono essere utilizzati. Un nodo viene selezionato nella Palette e trascinato sul workspace per essere posizionato.
- **Sidebar:** contiene una serie di tools aggiuntivi che possono essere utilizzati dall'utente.

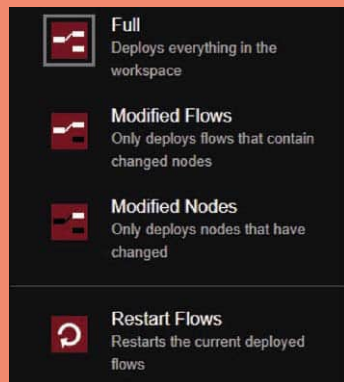


Fig. 8 - Differenti opzioni di deploy



Fig. 9 - Footer del workspace con la Navigation view attiva.

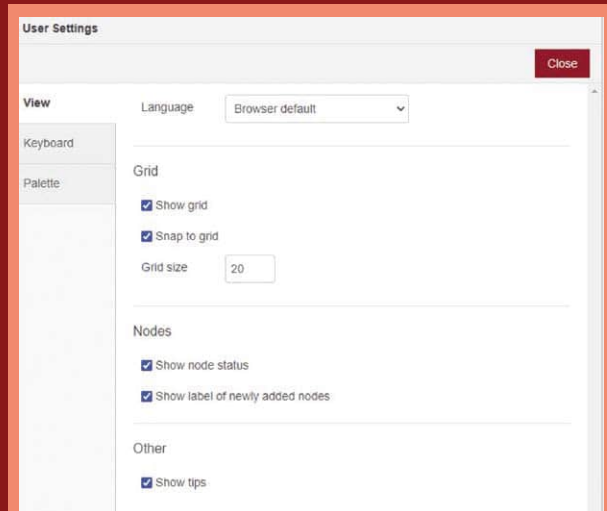


Fig. 10 - User Settings.

Analizziamo questi elementi più nel dettaglio per comprenderne meglio alcune caratteristiche:

Header

Come abbiamo visto, l'header contiene essenzialmente il menu principale, accessibile cliccando sull'icona con tre barre orizzontali ed il pulsante di deploy. Torneremo sul menu principale in seguito nelle puntate successive per illustrarne l'utilizzo. Per quanto riguarda invece il pulsante di deploy è importante sottolineare che la funzione di questo pulsante serve per rendere effettive le modifiche che abbiamo eseguito sul flow; sostanzialmente tutte le volte che modifichiamo un flow le informazioni vengono accumulate sul client ma non trasferite sul server che gira sulla Raspberry. Per far sì che il flow lato server venga aggiornato deve appunto essere premuto il pulsante di deploy, che si colora di rosso se il flow ha delle modifiche di cui non è ancora stato fatto il deploy. Questo pulsante permette 3 differenti modalità di deploy (come evidenziato in Fig. 8):

- **Full:** viene eseguito il deploy di tutti flows sul workspace
- **Modified Flows:** viene eseguito il deploy solo dei flow modificati
- **Modified Nodes:** viene eseguito il deploy solo dei nodi modificati

Workspace

Il workspace è la principale area di lavoro di Node-RED, dove i flows sono sviluppati prendendo nodi dalla Palette e collegandoli insieme. Il workspace ha una serie di tab in alto, corrispondenti ai vari flows che sono stati creati ed aperti. Il footer del workspace ha alcuni tool di ausilio alla navigazione dei flows (Fig. 9). In particolare sono presenti pulsanti per eseguire zoom e de-zoom nel flow e per ripristinare il valore di zoom di default. Inoltre è disponibile un pulsante per

attivare la navigazione del flow, se questo dovesse essere talmente grande da non poter essere visualizzato per intero nella finestra.

È inoltre possibile seguire alcune customizzazioni del workspace tramite l'opzione "settings" del menu principale, come visibile in Fig. 10.

Palette

La Palette contiene la lista di tutti i nodi installati e disponibili all'uso. I nodi sono organizzati in categorie (ad esempio alcune delle categorie di base sono common, output e functions). Cliccando l'header di una categoria è possibile comprimere o espanderne la visualizzazione (inoltre due pulsanti presenti al fondo della palette, con il simbolo di una doppia freccia verso l'alto o verso il basso possono essere utilizzati per comprimere ed espandere tutte le categorie). Nella parte alta della palette è presente anche un input testuale che permette di filtrare la lista dei nodi per nome.

Sidebar

Infine la sidebar è una barra laterale che contiene una serie di tools aggiuntivi a disposizione dell'utente. I tool di base sono i seguenti:

- **Information:** permette di visualizzare informazioni sui nodi.
- **Help:** permette di visualizzare l'Help dei nodi.
- **Debug:** permette di visualizzare i messaggi passati al nodo Debug.
- **Configuration Nodes:** permette di gestire i Configuration Nodes (un particolare tipo di nodo non visualizzato sul workspace).
- **Context Data:** permette di analizzare il contenuto dei context, un elemento che non è stato ancora presentato, e che verrà analizzato nella prossima puntata.

Node.JS

Node.js è un runtime system open source multiplatforma orientato agli eventi per l'esecuzione di codice JavaScript, costruito sul motore JavaScript V8 di Google Chrome ed originariamente creato da Ryan Dahl nel 2009. Normalmente gli script JavaScript vengono utilizzati lato client, generalmente incorporati all'interno dell'HTML di una pagina web, venendo interpretati da un engine incorporato direttamente all'interno di un Browser. Node.js consente invece di utilizzare JavaScript anche per scrivere codice da eseguire lato server, ad esempio per la produzione del contenuto delle pagine web dinamiche prima che la pagina venga inviata al Browser dell'utente. Node.js in questo modo permette di implementare il cosiddetto paradigma "JavaScript everywhere" (JavaScript ovunque), unificando lo sviluppo di applicazioni Web intorno ad un unico linguaggio di programmazione (JavaScript).

Node.js ha un'architettura orientata agli eventi che rende possibile l'I/O asincrono. Questo design punta ad ottimizzare il throughput e la scalabilità nelle applicazioni web con molte operazioni di input/output, è inoltre ottimo per applicazioni web real-time (ad esempio programmi di comunicazione in tempo reale o browser game).

Node.js è un progetto di sviluppo open source distribuito gestito dalla Node.js Foundation e facilitato tramite il programma di progetti collaborativi della Linux Foundation.

Le aziende che supportano il programma includono GoDaddy, Groupon, IBM, LinkedIn, Microsoft, Netflix, PayPal, Rakuten, SAP, Voxer, Walmart e Yahoo!

Inoltre alcuni nodi contribuiscono alla sidebar aggiungendo pannelli specifici (è il caso della Node-RED Dashboard, che vedremo in una delle puntate successive).

USARE NODE-RED SU RASPBERRY PI

Ora che abbiamo introdotto i concetti base di Node-RED e abbiamo presentato l'editor, passiamo al primo semplice esempio pratico, in modo da evidenziare la semplicità che Node-RED consente di raggiungere nello sviluppo delle applicazioni. Ciò che vogliamo realizzare è il classico lampeggio di un LED, pilotandolo da un GPIO della Raspberry alla frequenza di 1Hz. Per realizzare questo semplice esempio ci bastano 3 nodi:

- un nodo Inject;
- un nodo Trigger;
- un nodo Rpi – Gpio Out.

Iniziamo posizionando il nodo Inject, prelevandolo dalla Palette all'interno della categoria common. Il nodo Inject (Fig. 11) è utilizzato per iniettare dati sul flow, di diverso tipo e con varie modalità. Noi lo utilizzeremo per generare un inject periodico che fornisca la periodicità di esecuzione al nostro flow.

Una volta posizionato il nodo va configurato, per dargli il comportamento desiderato. In questo caso noi vogliamo che il nodo Inject fornisca la periodicità. Per configurarlo dobbiamo cliccare due volte sul nodo con il tasto sinistro del mouse. Una volta eseguita questa operazione si aprirà



Fig. 11 - Nodo Inject.

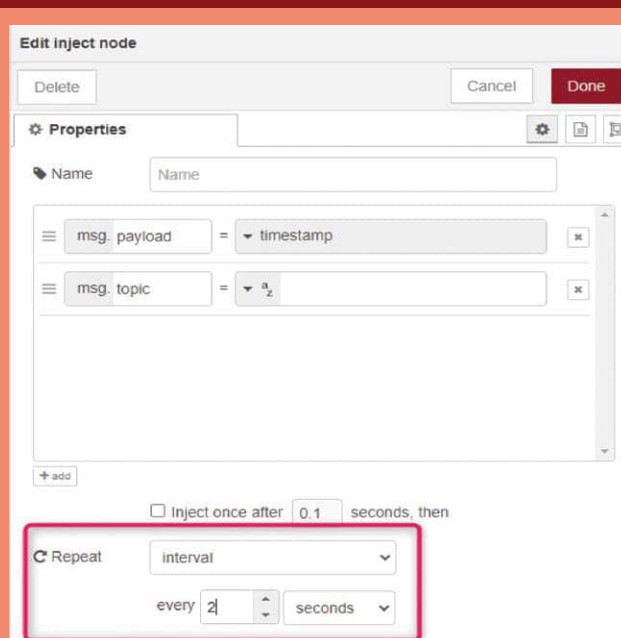


Fig. 12 - Configurazione del Nodo Inject.



Fig. 13 - Nodo Trigger.

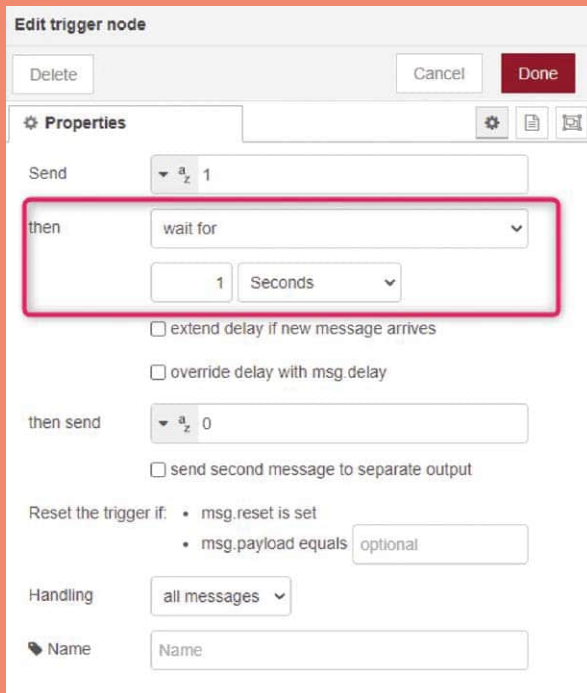


Fig. 14 - Configurazione del nodo Trigger

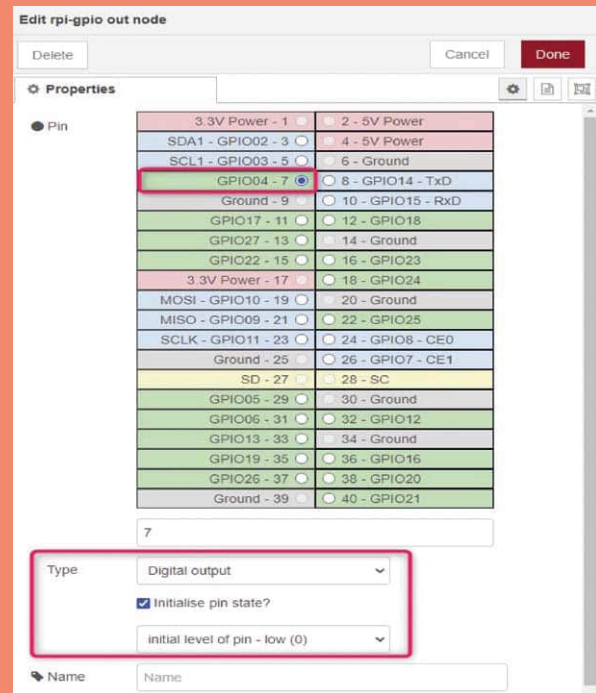


Fig. 16 - Configurazione del nodo rpi - gpio out

la finestra di Fig. 12, dalla quale è possibile configurare il nodo Inject. In questo particolare caso a noi basta configurare l'opzione Repeat, presente in basso, impostandola su Interval e indicando 2 secondi come intervallo. Non ci interessa modificare il Payload del nodo, che in questo caso è un timestamp, perché non lo utilizzeremo per pilotare direttamente il LED, ma solo per fornire la periodicità di attivazione. Una volta fatto clicchiamo su Done per chiudere. Ora possiamo inserire il nodo trigger, prelevandolo dalla categoria functions della Palette (Fig. 13). La funzione del nodo trigger è quella di inviare un messaggio con un determinato payload quando attivato e, dopo

un lasso di tempo configurato dall'utente, un secondo messaggio, sempre con un determinato payload. Questo è il nodo che noi utilizzeremo per generare l'informazione di controllo per il LED (prima on e dopo un secondo off). Apriamo la finestra di configurazione del nodo e modifichiamo il delay tra il primo e il secondo messaggio impostandolo ad 1 secondo (Fig. 14). In questo modo il nodo invierà, quando attivato, ossia alla ricezione di un messaggio in ingresso (il messaggio inviato dal nodo inject ogni 2 secondi), prima un messaggio contenente "1" e poi un secondo messaggio contenente "0". In questo modo abbiamo generato l'invio di un messaggio contenente "1" e dopo un secondo di un messaggio contenente "0" ogni 2 secondi, ossia abbiamo una serie di messaggi che rappresentano il "lampeggio" del nostro LED. Non ci rimane altro da fare che inserire un nodo che piloti direttamente il pin di GPIO cui vogliamo collegare il nostro LED.

Il nodo che fa al caso nostro è il nodo rpi - gpio out, presente nella categoria Raspberry Pi (Fig. 15). Posizioniamo il nodo sul workspace ed apriamone la configurazione. Qui dobbiamo semplicemente selezionare il GPIO che vogliamo utilizzare (il GPIO 4 nell'esempio, ma può essere utilizzato qualsiasi GPIO adatto allo scopo) e il tipo di gestione (campo Type: Digital output) e, opzionalmente, se vogliamo inizializzare il pin ad un certo livello logico, come illustrato in Fig. 16.

A questo punto non ci rimane altro da fare che creare i collegamenti. La creazione dei collegamenti è molto semplice,



Fig. 15 - Nodo rpi - gpio out.

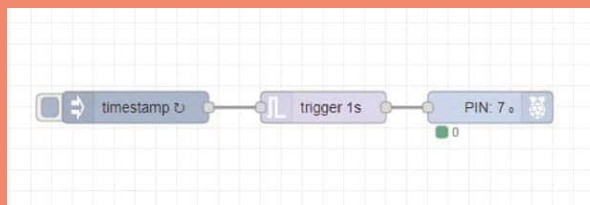


Fig. 17 - Flow per il lampeggio di un LED a 1Hz.

è sufficiente posizionarsi con il puntatore del mouse su una porta di un nodo cliccare e mantenere premuto il tasto sinistro, trascinare il puntatore sulla porta che si desidera collegare e rilasciare. Colleghiamo quindi l'output del nodo inject all'input del nodo trigger e l'output del nodo trigger all'input del nodo rpi - gpio out, come illustrato in Fig. 17. A questo punto possiamo premere il tasto deploy per mandare in esecuzione il nostro flow. Si noti che il nodo rpi-gpio out fornisce una indicazione dello stato fisico del pin (icona

verde nell'angolo in basso a sinistra del nodo, con raffigurato il valore logico assunto dalla linea di I/O che stiamo utilizzando). In Fig. 18 è rappresentato lo schema delle connessioni da utilizzare, usando una breadboard, un LED e una resistenza da 1kΩ, più dei normali cavetti di collegamento da breadboard. Per i nostri esperimenti abbiamo utilizzato una Raspberry Pi 4 Model B, ma sostanzialmente qualsiasi modello di raspberry è utilizzabile (per maggiori dettagli sul nodo RPI gpio è possibile consultare la relativa pagina sul sito di Node-RED: <https://flows.nodered.org/node/node-red-node-pi-gpio>).

CONCLUSIONI

Siamo arrivati alla conclusione di questa prima puntata di questo corso Node-RED; in questa puntata abbiamo spiegato cos'è Node-RED ed il principio della programmazione flow-based. Inoltre abbiamo iniziato a prendere confidenza con l'editor e visto i primi semplici esempi su Raspberry Pi. Nella prossima puntata inizieremo ad esaminare i più importanti nodi di base e vedremo alcune particolarità dell'ambiente, come ad esempio i context. □

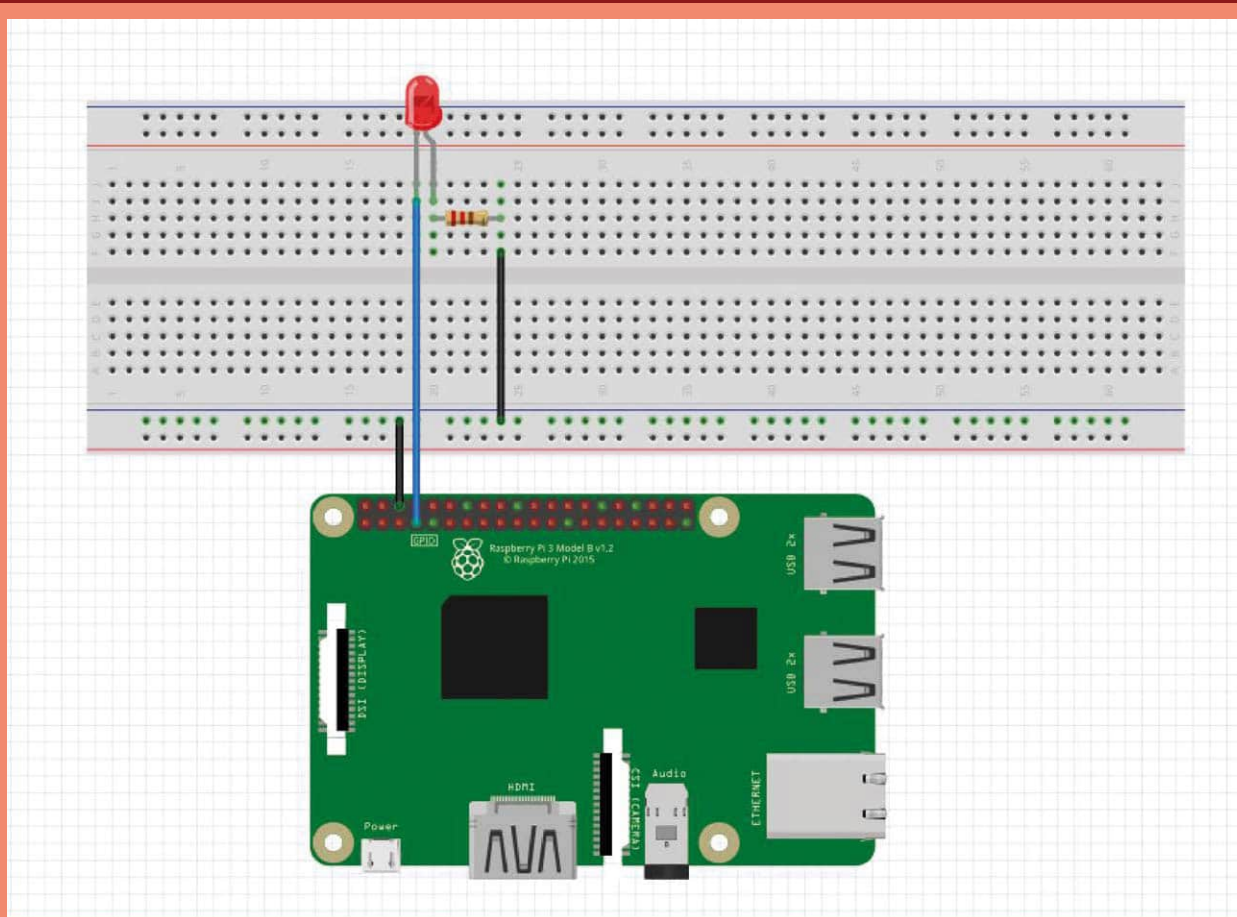


Fig. 18 - Schema dei collegamenti per il progetto di esempio.

Organoidi 3D cresciuti nello spazio

Con il prossimo volo di rifornimento alla Stazione Spaziale Internazionale (ISS), lo Space Hub dell'Università di Zurigo (UZH) e Airbus Defence and Space portano nello spazio un esperimento destinato per far avanzare ulteriormente la produzione industriale di tessuto umano in condizioni di gravità zero. Il progetto congiunto si chiama "Organoids 3D in the space" nasce da Oliver Ullrich e Cora Thiel, pionieri nella ricerca sull'influenza della gravità sulle cellule umane. Il team di Airbus Innovations sta sviluppando l'hardware e fornendo

l'accesso all'ISS. Dall'idea al primo test di produzione nello spazio, sono passati solo tre anni per completare fasi di test e procedure di selezione.

Oliver Ullrich, professore di anatomia all'UZH, la biologa Cora Thiel e Airbus stanno usando la microgravità nello spazio per far crescere tessuti tridimensionali simili a organi - i cosiddetti organoidi - da cellule staminali umane adulte. Sulla Terra, gli organoidi tridimensionali non possono essere prodotti senza sostenere gli scheletri a causa della gravità. Gli organoidi 3D sono



di grande interesse per le aziende farmaceutiche, perché consentono di condurre gli studi tossicologici direttamente sul tessuto umano risparmiando gli animali. Gli organoidi cresciuti dalle cellule staminali dei pazienti potrebbero anche essere utilizzati per la terapia sostitutiva dei tessuti per gli organi danneggiati, giacché il nume-

Accumulo gravitazionale: la massa prende e rilascia elettricità

Sollevare, impilare e riportare a terra blocchi di cemento: questo farà il sistema di stoccaggio stazionario ideato dalla startup svizzera Energy Vault, che propone una versione originale del concetto di "accumulo gravitazionale". Ispirato ai principi dell'idroelettrico a pompaggio, l'impianto è composto da una gru in acciaio a sei braccia, alta quasi 120 metri e da blocchi di 35 tonnellate, costituiti da materiale di scarto locale: dai rifiuti altrimenti destinati in discarica ai residui dell'attività mineraria.

Controllata attraverso un potente software, la gru dovrebbe sollevare i blocchi impilandoli uno sull'altro grazie all'energia elettrica prodotta da eolico e fotovoltaico nelle ore di bassa domanda; per rilasciare l'energia immagazzinata, li riporterebbe in basso sfruttando la gravità (ma in maniera controllata) e convertendo così l'energia cinetica dei blocchi in discesa in elettricità. Insomma, sfrutta elettricità a basso costo

ro di organi donati non copre la domanda mondiale. La ricerca condotta nel marzo 2020, quando 250 provette contenenti cellule staminali umane hanno trascorso un mese sulla ISS, ha avuto successo; invece le colture create sulla Terra, in normali condizioni di gravità, hanno mostrato una differenziazione cellulare nulla o solo minima.

Il materiale campione tornerà sulla Terra all'inizio di ottobre. I primi risultati del test sono attesi da novembre.

www.airbus.com

per produrre energia potenziale stivata nelle masse sollevate e quando serve corrente elettrica nelle ore di maggior domanda e quindi di prezzo più elevato, sfrutta l'energia cinetica rilasciata dalla discesa dei blocchi per produrre elettricità mediante quegli stessi motori utilizzati per il sollevamento. Per rendere il meccanismo il più efficiente possibile sono stati implementati algoritmi che calibrano e controllano i cicli di carica e scarica.

Il sistema è progettato per essere efficiente in termini di costi, affidabile, completamente sicuro da utilizzare e sostenibile dal punto di vista ambientale; inoltre è sostenuto da una scienza dei materiali all'avanguardia e da tecnologie e software proprietari.

La nuova piattaforma EVx, lanciata ad aprile 2021, introduce sensibili miglioramenti nel sistema, a partire da un'efficienza roundtrip (Rapporto fra l'energia ceduta nella fase di scarica e l'energia assorbita) dell'80-85%, una vita tecnica di oltre 35 anni e una progettazione flessibile e modulare con un'altezza inferiore del 45% rispetto al primo modello.

L'idea dell'azienda svizzera è iniziare a distribuire l'impianto di accumulo gravitazionale negli USA durante il quarto trimestre del 2021, puntando al mercato globale nel 2022.

<https://energyvault.com/>

Reti neurali per la difesa missilistica ispirate dalla libellula

Nel cervello umano, 86 miliardi di neuroni lavorano in parallelo, elaborando gli input dei sensi e dei ricordi, dando luogo alla cognizione umana. I cervelli di altri esseri viventi sono meno capaci, ma spesso mostrano attitudini innate per compiti particolari. Di recente la ricerca guarda con attenzione tali sistemi nervosi specializzati, come modello per l'intelligenza artificiale. Presso i Sandia National Laboratories di Albuquerque (laboratori di sicurezza nazionale U.S.A.) si sta studiando il cervello della libellula, che ispira la progettazione di sistemi informatici per l'intercettazione dei missili. Sfruttando la velocità, la semplicità e l'efficienza del sistema nervoso della libellula, si punta a sviluppare computer che eseguano queste funzioni più velocemente e con una frazione della potenza consumata dai sistemi convenzionali. Questo è importante perché sebbene le reti neurali possono simulare e superare le potenzialità del cervello umano, richiedono enormi potenze di elaborazione, nonché supercomputer più veloci e risorse per supportarli. Stime recenti suggeriscono che le

emissioni di carbonio derivanti dallo sviluppo e dall'addestramento di un algoritmo di intelligenza artificiale sono maggiori di quelle prodotte da quattro automobili nel corso della loro vita. La libellula impiega circa 50 millisecondi per iniziare a rispondere alla manovra evasiva di una preda; tolti 10 ms affinché le cellule dell'occhio rilevino e trasmettano informazioni sulla preda e altri 5 ms affinché i muscoli correggano la traiettoria, il circuito neurale impiega solo 35 ms per effettuare i suoi calcoli. Dato che in genere un singolo neurone impiega almeno 10 ms per integrare gli input, la rete neurale sottostante può essere profonda almeno tre strati. Le libellule catturano fino al 95% delle prede che inseguono; un'ottima percentuale di affidabilità se trasposta su un sistema che deve inseguire e abbattere un missile anche se questo tenta manovre evasive. Non a caso per decenni le agenzie statunitensi hanno sperimentato progetti ispirati alla libellula per i droni di sorveglianza e i sistemi anti-missile, relizzabili con limitata potenza di calcolo, eseguendo poche ma precise e rapide operazioni.

https://newsreleases.sandia.gov/dragonfly_defense/



Potenza di calcolo per la mobilità di domani

Guardando ai veicoli del futuro, sempre più automatizzati e autonomi, ZF punta sull'Intelligenza artificiale per le proprie unità di gestione delle auto. L'ultima generazione del supercomputer ZF ProAI offre potenza di calcolo personalizzata per qualsiasi veicolo e livello di automazione.

Il cuore delle nuove architetture dei veicoli E/E è costituito da computer ad alte prestazioni che possono essere utilizzati come controller centrali, di dominio o di zona. Il supercomputer ZF ProAI è adatto a tutte le fasi della guida automatizzata o autonoma. Le sue funzionalità sono ottimizzate per il deep learning. Il computer fornisce la fusione a 360° basata su GPU di tutti i dati dei sensori disponibili, inclusi dati di misurazione ambientale da telecamere, radar, LiDAR e modelli audio. Il tutto in 24x14x5 cm, compreso il sistema di raffreddamento (passivo, ad aria e a liquido a seconda delle prestazioni desiderate).



Il supercomputer può essere equipaggiato con diverse varianti di "System-on-Chip" (SoC), ovvero con processori di diversi produttori, a seconda delle esigenze del cliente. Può essere utilizzato anche con software di ZF o di fornitori terzi. I connettori standardizzati e la possibilità di collegare tra loro più unità ZF ProAI lo rendono flessibile per l'utilizzo e l'installazione in qualsiasi tipo di veicolo.

Il computer ad alte prestazioni di ZF offre un'enorme potenza di calcolo: da 20 a 1.000 teraOPS (fino a 1.000 trilioni, o un quadrilione, di operazioni di calcolo al secondo). Il supercomputer è rivolto ad applicazioni software e architetture E/E, tramite un unico tipo di ECU. ZF produrrà il controller centrale ProAI su larga scala dal 2024, per autovetture e veicoli commerciali leggeri.

www.zf.com

Il casco smart che identifica l'ictus

Quando si verifica un ictus, più in fretta si perviene alla diagnosi e alla terapia, più è possibile limitare i danni. Idealmente, i pazienti dovrebbero essere diagnosticati e trattati entro la prima ora. Data una tempistica così stretta, numerosi team di ricerca hanno sviluppato caschi intelligenti portatili per diagnosticare l'ictus nei pazienti mentre vengono trasportati in ospedale. Molti caschi si basano sugli ultrasuoni per creare immagini del cervello, ma gli ultrasuoni richiedono personale

qualificato per interpretare correttamente le immagini e non penetrano nel cranio come le onde elettromagnetiche. Invece il sistema sviluppato presso l'Università di Genova, basato sulle onde elettromagnetiche, consente diagnosi più semplici e distinte per i due tipi di ictus, ischemico ed emorragico, i quali hanno proprietà dielettriche diverse.

Il casco è composto da antenne attivate selettivamente per dirigere le onde EM in tutto il cervello; vengono misurati i segnali di ritorno e un algoritmo di elaborazione del segnale avvisa se si è verificato un ictus e un altro determina il tipo, la dimensione e la posizione. I ricercatori hanno testato il casco simulando varie posizioni e dimensioni (tra 1 e 4 cm) all'interno del cervello. Ora si attendono gli studi clinici.

www.aem.diten.unige.it/drupal/news



L'auto fotovoltaica alimenta la rete

Sono Motors, azienda tedesca creatrice di Sion, l'auto elettrica alimentata da pannelli fotovoltaici, ha trasformato la sua vettura in una centrale elettrica semovente: grazie all'innovativa stazione di ricarica a parete (wall box) può non solo ricaricare la propria batteria ma anche erogare elettricità alla rete elettrica se i suoi pannelli sono al sole. Il suo wall box AC bidirezionale, primo al mondo, presenta un'efficienza del 70% superiore a quella delle stazioni di ricarica domestica sul mercato. Il wall box consente di utilizzare la Sion come centrale elettrica mobile e anche di reimmettere nella rete elettrica l'elettricità accumulata; supporta una potenza di 11 kW tramite il cavo di ricarica integrato di tipo 2. L'elevata efficienza si deve al fatto che essendo in corrente alternata, il wall-box è più semplice, perché la conversione da corrente continua a corrente alternata avviene all'interno della Sion e non nel wallbox. Un sistema di gestione dell'energia domestica (HEMS) controlla l'energia in ricarica: il wall box comunica con l'HEMS tramite vari protocolli come EEBUS, Modbus o OCPP. L'app Sono e l'infotainment Sono possono essere utilizzati anche per impostare il livello della batteria del veicolo al quale è possibile terminare la scarica. Il wall box può caricare la batteria LFP da 54 kWh in un massimo di 5 ore, fornendole un'autonomia fino a 305 km. Inoltre, Sion può essere ricaricata presso stazioni di ricarica rapida fino a 75 kW (DC) tramite l'interfaccia CCS integrata, ovvero a un livello di carica dell'80% in circa 35 minuti.

<https://sonomotors.com/>

NUOVI KIT PER VIDEOGIOCHI ARCADE con Raspberry Pi 4 !



KIT CABINATO ARCADE 1 GIOCATORE CON MONITOR



€ 299,00

Cod. **GAMECABINET**

CONSOLE 2 GIOCATORI PER VIDEOGIOCHI ARCADE



€ 239,00

Cod. **GAMECONSOLE**



Mini postazione per videogiochi Arcade come quelle degli anni '80 e '90! Compatibile con Raspberry Pi 4 tipo B (versione da 2-4-8 GB), utilizza come software RetroPie (aggiornato all'ultima versione), con il quale è possibile emulare vari tipi di console, tra cui: Nintendo, Game Boy, MAME, Sega Master System, Amiga, Commodore, ecc.... Raspberry Pi 4 tipo B e la scheda SD contenente il software RetroPie sono acquistabili separatamente. Grazie alle dimensioni contenute è possibile utilizzarlo ovunque!

Console per videogiochi Arcade con pannello in plexiglass serigrafato. Compatibile con la Raspberry Pi 4 tipo B (Versione da 2-4-8 GB), utilizza come software RetroPie (aggiornato all'ultima versione), con il quale è possibile emulare vari tipi di console, tra cui: Nintendo, Game Boy, MAME, Sega Master System, Amiga, Commodore, ecc.... Raspberry Pi 4 tipo B, la scheda SD contenente il software RetroPie e il monitor sono acquistabili separatamente.

€ 72,00

Cod. **RPI4-4GB**

**RASPBERRY PI 4
TIPO B CON 4GB
DI MEMORIA**



€ 14,90

Cod. **RETROPIESD32GB**

**SDCARD DA 32GB
CON SISTEMA
RETROPIE**



Prezzi IVA inclusa

**FUTURA
ELETTRONICA®**

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

Caratteristiche tecniche
e vendita on-line su:

www.futurashop.it

Purificatore d'aria a raggi ultravioletti

VERSIONE
DA PARETE

€179,00

Cod. MICROUVAIR-P

Alimentazione: 12 Vcc

Potenza assorbita: @ 3.6 W

Durata sorgente UV: @20000 ore

Capacità sterilizzazione: 60 m³/h

Dimensioni: altezza 25 cm x 10.5 cm x 7 cm

È un sistema di sterilizzazione dell'aria che assicura l'eliminazione di virus e batteri.

Riduce il rischio di contaminazione delle persone all'interno degli uffici.

Rende gli ambienti batteriologicamente sani eliminando anche le spore.

È in grado di effettuare un trattamento di disinfezione dell'aria in ambienti fino a 60 m³.

Può funzionare continuativamente anche in presenza di persone.

Abbatte il 99% della carica batterica aerea dopo circa un'ora di funzionamento.



MICROUVAIR è un sistema di sanificazione aria con ventilazione forzata a ciclo chiuso basato su led UVC. L'aria in ingresso viene purificata attraverso un filtro ad alta densità (30 PPI), quindi viene sterilizzata tramite luce ultravioletta emessa da led UVC in grado di effettuare la massima azione germicida. L'aria sanificata viene quindi filtrata ulteriormente da un filtro meccanico ad alta densità e quindi rimessa nell'ambiente.



VERSIONE
DA TAVOLO

€199,00

Cod. MICROUVAIR-T



Prezzi IVA inclusa.

SANIFICATORE UVC PER OGGETTI

Sanificatore automatico con LED UVC, particolarmente indicato in tutte le situazioni dove si rende necessario igienizzare, sanificare e comunque disinfettare oggetti potenzialmente depositari di particelle infettive (virus, batteri, ecc.) come mascherine chirurgiche, utensili, occhiali, chiavi, ecc. È costituito da un contenitore in plexiglass con un coperchio incernierato all'interno del quale si trova una strip a LED UV-C con quattro LED SMD UV-C tipo 3535 da 5 mW ciascuno con lunghezza d'onda della radiazione emessa nella banda dei 265 e i 278 nm, gamma ideale per sterilizzare e sanificare superfici e oggetti. Il set contiene tutte le parti in plexiglass, le parti stampate in 3D, l'elettronica e l'alimentatore.



€80,00

Cod. UVCSAN

