

Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXVII - n. 257 - SETTEMBRE 2021

Bobina di Tesla su circuito stampato

■ **Generatore di funzioni JDS6600-60M**

■ **Stampiamo con un ESP8266**

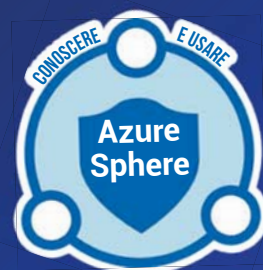
■ **MiniUPS per Raspberry Pi**

■ **Fattore di smorzamento del finale audio**

■ **Allarme basato su AI**

Logger GPS con uCHIP

Smart Mirror con Alexa



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.

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)

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

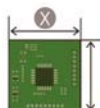
1 Layer

2 Layers

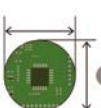
4 Layers

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



cod. PCBPRODUCTION

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.

Prezzi IVA inclusa.

www.futurashop.it

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

**FUTURA
 ELETTRONICA®**

EDITORIALE

Direttore Responsabile

Boris Landoni

@ boris.landoni@elettronica.in

Redazione

Stefano Garavaglia, Paolo Gaspari,
Davide Scullino, Gabriele Daghetta

@ redazione@elettronica.in

Grafica

Alessia Sfulcini

@ alessia.sfulcini@elettronica.in

Ufficio Pubblicità

Monica Premoli (0331-752668)

@ monica.premoli@elettronica.in

Ufficio Abbonamenti

Tel. 0331-752668

@ abbonati@elettronica.in

Direzione, Redazione, Pubblicità

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

Tel. 0331-752668

Abbonamenti

Annuo 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.in/abbonamenti/>

Distribuzione per l'Italia

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

via Bettola 18 - 20092 Cinisello Balsamo (MI)

Tel. 02-660301 Fax 02-66030320

Stampa

ROTO3 Spa - Via Turbigo, 11/b
20022 CASTANO PRIMO (MI)Tiratura
18.000 copieMensile
associato all'USPI,
Unione Stampa
Periodica ItalianaRivista mensile registrata presso il Tribunale di Milano
con il n. 245 il 03/05/1995.

Prezzo di copertina Euro 6,00.

Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili
sul sito della rivista al prezzo di Euro 6,00.Poste Italiane Spa - Spedizione in abbonamento Postale - D.L.
353/2003 (conv. in L. 27/02/2004) art. 1 comma 1 - DCB Milano.Futura Group srl è iscritta al Registro Operatori della
Comunicazione n. 23650 del 02/07/2013.Impaginazione ed immagini sono realizzati in DeskTop Publishing
con programmi Adobe InDesign e Adobe Photoshop per Windows.
Tutti i contenuti della Rivista sono protetti da Copyright.Ne è vietata la riproduzione, anche parziale, la traduzione e più in
generale la diffusione con qualsiasi mezzo senza l'autorizzazione
scritta da parte dell'Editore. I circuiti, il firmware ed il software
descritti sulla Rivista possono essere realizzati solo per uso
personale, ne è proibito lo sfruttamento a carattere commerciale
e industriale. Tutti possono collaborare con Elettronica In. L'invio
di articoli, materiale redazionale, programmi, traduzioni, ecc.
implica da parte del Collaboratore l'accettazione dei compensi e
delle condizioni stabilite dall'Editore (www.elettronica.in/ase.pdf).
Manoscritti, disegni e foto non richiesti non verranno in alcun caso
restituiti. L'utilizzo dei progetti e dei programmi pubblicati non
comporta alcuna responsabilità da parte della Società Editrice.

© 2021 Futura Group srl

Futura
Group
Edizioni

I buoni propositi...

Settembre è il mese della ripresa delle attività: terminano le vacanze estive, dalle quali torniamo ristorati, carichi di buone intenzioni e prodighi di buoni propositi. Ai classici "perdere peso" e "imparare l'inglese", quest'anno dobbiamo aggiungere anche un ambizioso "salvare il pianeta". Infatti il 14 luglio la Commissione Europea (e quindi tutti noi) si è data l'importante obiettivo di fare dell'Europa il primo continente a impatto

climatico zero entro il 2050 e quindi di tradurre il Green Deal europeo in una realtà concreta.

La timeline vede una serie di traguardi per il 2030 (sì, esatto: tra meno di 9 anni), in particolare si punta alla riduzione del 55% delle emissioni delle automobili,

per poi arrivare a vendere, per il 2035, solamente vetture con zero emissioni. 55% è anche la quota di riduzione prevista per le emissioni di gas a effetto serra entro il 2030, ottenibile portando al 40% la percentuale di energia rinnovabile nel mix energetico dell'UE.

A questo dobbiamo aggiungere l'ottimizzazione delle classi energetiche degli edifici, il ripristino della vegetazione finalizzato all'assorbimento della CO₂, una cooperazione internazionale per la lotta contro i cambiamenti climatici e molti altri, più "piccoli", obiettivi.

Il 2030 non è lontano e tutti dobbiamo fare la nostra parte, iniziando ora, che dalle vacanze abbiamo attinto nuova energia. Nel perseguimento di tali obiettivi, l'Elettronica è in qualche modo protagonista: impararla, praticarla e usarla per elevare l'efficienza di ogni sistema è il nostro contributo al Green Deal!

Boris Landoni

@ boris.landoni@elettronica.in

ON LINE

@ elettronica.in

SI RINGRAZIA

Futura Elettronica
Microchip
Mouser

In copertina:
La bobina di Tesla è sempre affascinante. In questo numero la proponiamo in versione da circuito stampato.

Contenuti

01

RUBRICHE

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

Hi-Tech

Logger GPS
con uCHIP



47

19

ARTICOLI

IN VETRINA

19 Generatore di Funzioni JDS6600-60M

Generatore di funzioni basato sulla tecnologia DDS, indispensabile per hobbisti e riparatori.

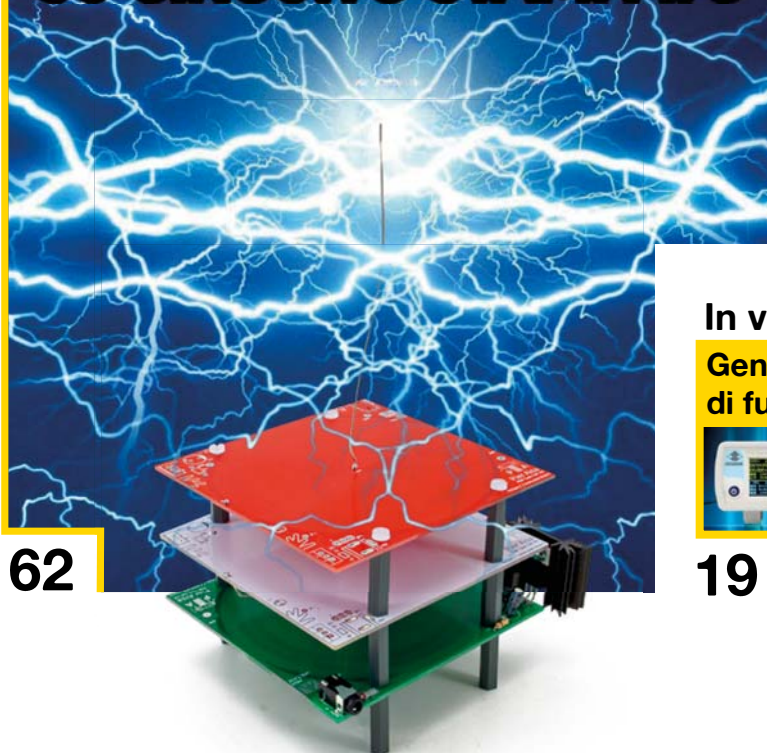
RASPBERRY PI

33 Smart Mirror con Alexa

Specchio Smart basato su Amazon Alexa e Raspberry Pi, controllato da comandi vocali o gesti della mano può diventare l'interfaccia utente di una smart-home. Prima puntata.

Progetto di Copertina

BOBINA DI TESLA SU CIRCUITO STAMPATO



62

In vetrina
Generatore
di funzioni



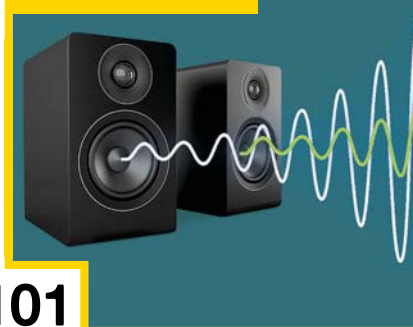
19

Raspberry Pi

Smart Mirror con Alexa



33

Networking**Stampiamo on-line
tramite smartphone****73****Raspberry Pi****MiniUPS: continuità
per il lampone****89****Didattica****Corso
Azure Sphere****117****Audio****Fattore di smorzamento
del finale audio****101****Sicurezza****Allarme basato su AI****107****HI-TECH****47 Logger GPS con uCHIP**

Crea log delle posizioni rilevate da un ricevitore GPS e ne elabora i dati per calcolare la distanza e la rotta dalla posizione da cui si è partiti, in modo da poter ritornare facilmente al luogo di partenza.

LABORATORIO**62 Bobina di Tesla su circuito stampato**

Costruiamo una bobina di tesla realizzandone l'avvolgimento secondario su PCB, così da risparmiarci l'onere dell'avvolgimento manuale.

NETWORKING**73 Stampiamo on-line
tramite smartphone**

Controlliamo un dispositivo seriale con il modulo WiFi ESP8266. I comandi saranno inviati da remoto attraverso il protocollo MQTT.

RASPBERRY PI**89 MiniUPS: continuità per il lampone**

Piccolo gruppo di continuità per Raspberry Pi in grado di mantenere accesa la board in caso di blackout. Seconda ed ultima puntata.

AUDIO**101 Fattore di smorzamento
del finale audio**

Scopriamo cos'è il fattore di smorzamento di un amplificatore audio e misuriamone l'entità con la scheda audio di un computer.

SICUREZZA**107 Allarme basato su AI**

Sistema anti-intrusione sensibile solo alle figure umane, che genera notifiche di allarme tramite instant messaging su Telegram.

DIDATTICA**117 Conoscere e usare Azure Sphere**

Utilizziamo il sensore MPU6050 incorporato nella nostra demoboard per misurare i dati dell'accelerometro e del giroscopio e visualizzarli in Azure IoT Central. Sesta e ultima puntata.

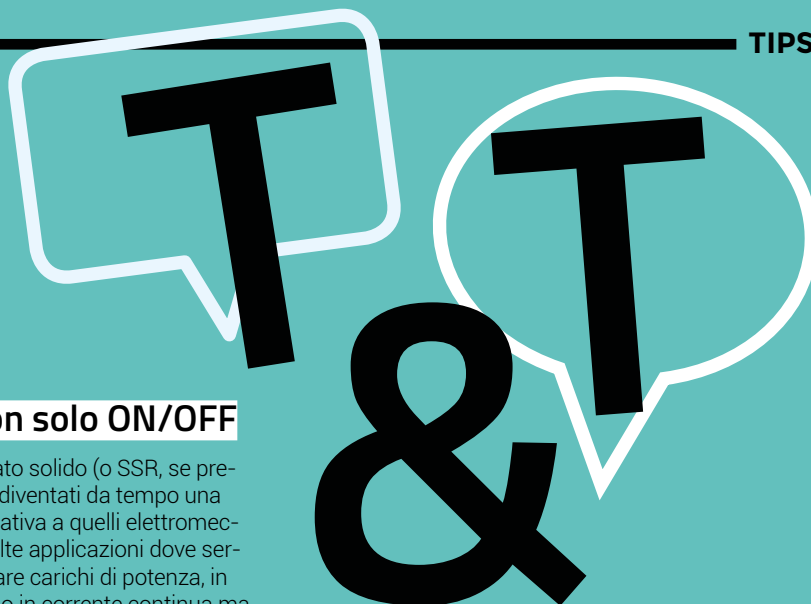
Un'offerta sempre più grande

La più ampia selezione di semiconduttori e componenti elettronici disponibili e pronti per la spedizione



[mouser.it](https://www.mouser.it)





SSR: non solo ON/OFF

I relé allo stato solido (o SSR, se preferite) sono diventati da tempo una valida alternativa a quelli elettromeccanici in molte applicazioni dove serva commutare carichi di potenza, in special modo in corrente continua ma non solo (ne esistono per l'alternata, formati da fotoaccoppiatori e TRIAC). Gli SSR (Solid State Relay) non solo permettono la commutazione senza consumo dei contatti né i limiti dei componenti elettromeccanici, tantomeno il rumore dell'equipaggio mobile e il consumo della bobina eccitatrice, ma hanno qualcosa in più sotto l'aspetto del tipo di controllo del carico; infatti di norma sono fatti per accendere e spegnere il circuito dell'utilizzatore, ma in molti casi la loro circuitazione è tale, soprattutto negli SSR per corrente continua, da consentire la modulazione PWM della potenza erogata, ossia la variazione del valore medio di tensione e quindi di corrente nel carico commutando rapidamente (comunque in bassa frequenza) per periodi variabili, giocando sul rapporto tra stato di ON e periodo della pulsazione. La massima frequenza di pilotaggio dell'ingresso del relé allo stato solido dipende dalle caratteristiche sia dello stadio d'ingresso, sia dello stadio d'uscita e dalla presenza di condensatori di filtro e comunque è almeno di qualche centinaio di hertz. La si può comunque dedurre dal tempo di risposta dichiarato dal costruttore: se tale tempo (T_{on} per l'attivazione e T_{off} per il ritorno in interdizione) è dell'ordine del millisecondo, verosimilmente si può pilotare l'ingresso a una frequenza 10 volte inferiore, quindi anche intorno ai 100 Hz.

Calcolo facile del dissipatore

Energy Harvesting è un termine che molti sperimentatori sono in

Tips & Tricks sono i nostri suggerimenti in pillole e i piccoli trucchi riguardanti sia l'hardware che il firmware, che sperimentatori, tecnici e progettisti elettronici vorrebbero conoscere per sviluppare meglio la propria passione: quella per l'Elettronica applicata. Ecco quelli che abbiamo preparato questo mese per voi.

difficoltà quando devono calcolare il dissipatore occorrente in un circuito di potenza che utilizza componenti a semiconduttore per commutare la corrente; del resto la difficoltà può starci perché si tratta di fare valutazioni termiche alle quali chi è abituato a dimensionare circuiti a transistor e

a ragionare coi bit, è poco avvezzo. Per semplificare la vita a chi deve progettare stadi del genere esistono strumenti on-line come quello disponibile sul sito www.engtools.online/heatsink.html, che propone una pagina web dove, in apposite caselle, inserire i dati principali del circuito; tali parametri

Con questa utilità è possibile calcolare le alette di raffreddamento per transistor, mosfet, diodi e resistenze di potenza. È possibile effettuare i seguenti calcoli:

- Progetto per ricavare il valore di resistenza R_{th} del dissipatore.
- Verifica per determinare le temperature di funzionamento.
- Determinare la potenza massima dissipabile per una determinata configurazione.

Se il dissipatore è impiegato per raffreddare anche altri componenti, nel campo P_i bisogna inserire la potenza complessiva dissipata dagli altri componenti.

Tipo di calcolo:

Dati noti del transistor:

Potenza massima transistor: P_{max} W @ T_c : °C

Temperatura massima giunzione: T_{jmax} °C

Temperatura massima ambiente: T_{amax} °C

Resistenza termica contenitore/dissipatore: R_{ch} °C/W

Numero transistor sullo stesso dissipatore: N

Potenza dissipata da ogni transistor: p W

↑ Pagina web www.engtools.online/heatsink.html per il calcolo rapido dei radiatori di calore per elettronica.



sono poi gli stessi da cui bisogna partire nel calcolare un dissipatore di calore, ossia la differenza di temperatura massima tra la giunzione (in questo caso è la T_{jmax} , ossia massima temperatura che il semiconduttore può raggiungere prima di danneggiarsi) e l'ambiente (T_a) dove il radiatore deve disperdere il calore, ma anche la potenza da smaltire. Infatti nel calcolo della resistenza termica (ossia la fatica che il calore fa ad uscire dal componente per disperdersi nell'ambiente) complessiva si tiene conto del salto termico e della potenza da dissipare, secondo la formula:

$$\theta_{ja} = \Delta t / W$$

dove θ_{ja} è la resistenza termica tra giunzione e ambiente, W la potenza da dissipare e Δt il salto termico.

Nel calcolo del dissipatore va tenuto conto della resistenza termica fra giunzione e contenitore (θ_{jc}) e fra contenitore e radiatore (θ_{cr}) che dipende dal tipo di accoppiamento termico (con o senza pasta, tramite foglietto isolatore ecc.).

I terminali dei componenti THT

Nella saldatura, i terminali dei componenti a montaggio tradizionale (a foro

passante o THT, che dir si voglia) hanno una loro importanza e una funzione specifica, per assolvere la quale presentano determinate forme e dimensioni; per questa ragione è importante non tagliarli prima di aver completato la saldatura, ovvero prima di aver stagnato piazzola e componente e lasciato raffreddare la lega saldante. Questo perché i reofori fanno anche da dissipatore di calore, raffreddando più velocemente la saldatura e disperdendo in fretta il calore accumulato nel componente, così da limitare il rischio di danneggiarlo per surriscaldamento. Se i terminali si lasciano lunghi durante la saldatura tale funzione viene assolta, mentre se si tagliano prematuramente occorre un po' più d'attenzione, perché il componente può restare caldo più a lungo del dovuto ed è maggiormente esposto a rischi. Quindi i terminali sono utili non solo per montare e collegare i componenti, ma anche per ottimizzarne la saldatura e la loro forma e consistenza non è casuale.

Arduino: quando la seriale dipende dal micro

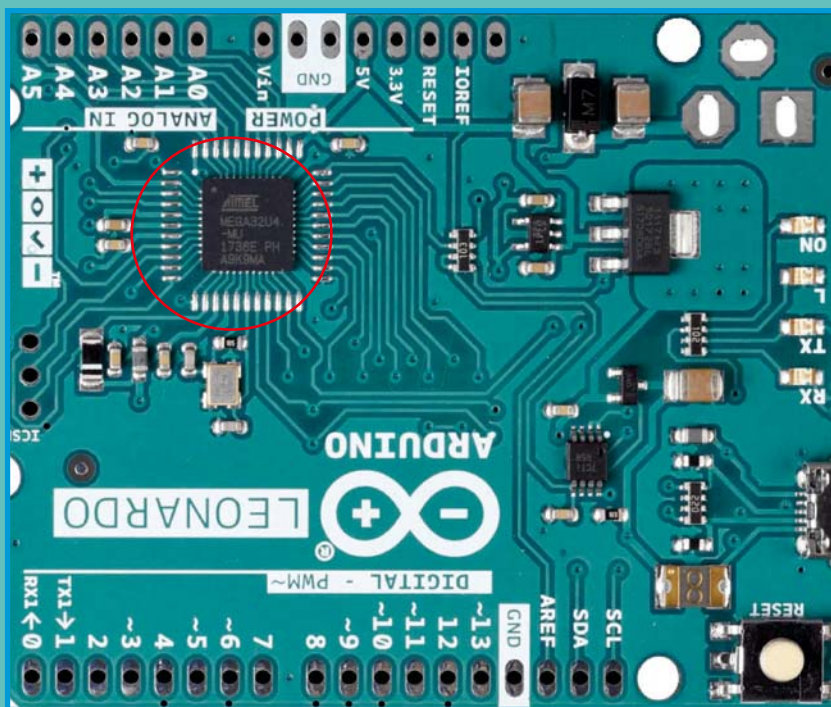
Sotto l'aspetto della connessione USB, le schede Arduino, che pure costituiscono uno standard, non sono tutte uguali, infatti sotto l'aspetto della programma-

zione, vale a dire del caricamento degli sketch, Arduino Leonardo e Micro, ma anche schede compatibili, sono diverse dalle altre schede Arduino.

Infatti le board basate sul microcontrollore Atmega32u4 e quindi quelle appena citate, diversamente dalle altre Arduino utilizzano un unico microcontrollore sia per eseguire gli sketch che per la comunicazione USB con il Personal Computer. La Uno e le altre schede utilizzano chip separati per queste due funzioni (ossia un microcontrollore Atmega per l'elaborazione e un transceiver USB/UART per la porta USB), il che significa che la connessione USB al computer rimane stabilita indipendentemente dallo stato del microcontrollore principale e da quale programma sta eseguendo, ovvero se è stato resettato. Combinando queste due funzioni in un unico processore, l'Atmega32u4, appunto, la Leonardo e le schede similari consentono una maggiore flessibilità nella comunicazione USB, però impongono delle limitazioni che bisogna conoscere.

Poiché tali schede non hanno un chip dedicato per gestire la comunicazione seriale, la porta seriale è virtuale: è una routine software, sia sul sistema operativo che sulla scheda stessa; insomma, qualcosa tipo quella implementata dalla libreria **newsoftserial** con cui si crea una seriale virtuale. Proprio come nei sistemi operativi Microsoft si crea un'istanza del driver della porta seriale quando viene connessa una periferica USB, la scheda Arduino Leonardo o la Micro creano un'istanza seriale ogni volta che viene eseguito il bootloader. Quando si collega una scheda del genere al Personal Computer, essa rappresenta un'istanza del driver Connected Device Class (CDC) dello standard USB, pertanto ogni volta che si resetta la scheda, la connessione seriale USB viene interrotta e poi ristabilita. La scheda scomparirà dall'elenco delle porte seriali e l'elenco verrà nuovamente enumerato.

Qualsiasi programma che ha una connessione seriale aperta su Leonardo/ Micro la perderà; questo contrariamente ad Arduino Uno, con la quale è possibile resettare il processore principale (l'ATmega328P) senza chiudere la connessione USB (che è gestita dal processore secondario ATmega8U2 o ATmega16U2). Questa differenza ha implicazioni per l'installazione, il caricamento e la comunicazione dei driver. A differenza della Uno, le schede Arduino basate su Atmega 32u4 non riavviano lo sketch quando si deve aprire una porta



↑ Il processore Atmega32u4 di alcune Arduino gestisce anche l'interfaccia USB.



seriale sul computer. Ciò significa che non saranno visibili i dati seriali che sono già stati inviati al computer dalla scheda, inclusi, ad esempio, la maggior parte dei dati inviati nella funzione **setup()**. Questa modifica significa che se stiamo utilizzando una qualsiasi delle istruzioni **Serial print()**, **println()** o **write()** nella nostra configurazione, questa non verrà visualizzata quando apri il monitor seriale. Per ovviare a questo si può verificare, prima di svolgere certe operazioni, se la porta seriale risulta aperta dopo aver chiamato **Serial.begin()** impartendo queste istruzioni:

```
Serial.begin(9600);

// while the serial stream is not
open, do nothing:

while (!Serial) ;
```

Il LED che da spento sembra acceso

Sarà capitato a qualcuno di voi di vedere dei faretto o lampadine a LED illuminati, pur quando sono stati appena spenti. Questo comportamento non deve far gridare al miracolo, ma è del tutto spiegabile considerando la tecnologia costruttiva dei LED bianchi; infatti, a differenza delle luci rossa, gialla, arancio, verde, blu e via di seguito, la luce bianca non viene ottenuta direttamente ma è il risultato della conversione fatta da uno strato di fosfori che riveste l'interno della lente o della cavità del chip. Più esattamente, il tipico diodo a luce bianca è in realtà un LED con emissione a luce blu, violetta o ultravioletta, che stimola il rivestimento di fosfori che avvolge la zona emittente della giunzione a produrre a sua volta fotoni che compongono la tinta bianca, calda o fredda in base ai materiali utilizzati.

Proprio il fatto che tali fosfori sono stimolati dalla luce, fa sì che spegnendo i LED e guardandolo al buio lo si noti debolmente illuminato, al pari di quanto avviene con le lancette e le tacche dei quadranti degli orologi tradizionali.

Il PID, questo sconosciuto...

Spesso quando si parla di controlli automatici, per esempio di temperatura, si

vede tra le caratteristiche degli apparati il termine PID: si tratta di un sistema di controllo ad anello chiuso che coniuga più tecniche per pervenire a un controllo ottimale, stabile e preciso.

Per comprendere il funzionamento del PID si può partire dallo spiegarne una componente, che è la regolazione proporzionale; infatti PID sta per Proporzionale Integrativo Derivativo. Nel controllo proporzionale, più il valore rilevato differisce da quello atteso, più energica è l'azione di avvicinamento: nel caso di un controllo di temperatura, più questa è minore del valore impostato, più energia si applica ad esempio al riscaldatore e viceversa. Sembrerebbe un ottimo metodo e lo sarebbe se non ci fosse l'inerzia (termica nel caso esemplificato o meccanica nel caso di un motore che deve raggiungere una posizione), la quale determina un'ondulazione attorno al valore atteso. Inoltre, il fatto di annullare la corrente una volta raggiunta la temperatura o posizione implica che questa non verrà mai raggiunta: ci si avvicinerà (in assenza di inerzia) all'infinito, senza mai raggiungerla. E più si sarà prossimi, più lentamente la regolazione interverrà, risultando peraltro sensibilissima ai disturbi esterni, visto che l'energia applicata per la correzione è nulla. Se il sistema si sta avvicinando rapidamente o lentamente al valore impostato, per il controllo proporzionale è irrilevante, perché valuta solo la differenza tra l'attuale e quello atteso. Il sistema PID risolve questo problema perché tiene conto di come avviene la variazione della posizione, aggiungendo un controllo Derivativo, vale dire Differenziale.

Questo fattore tiene in conto della rapidità con cui diminuisce l'errore (scarto tra valore atteso e attuale) nel tempo: se ci si avvicina rapidamente, la diminuzione è rapida e quindi occorre interrompere prima l'azione di controllo. Inserendo tale parametro è possibile compensare quasi totalmente i problemi dovuti all'inerzia meccanica o elettrica.

La parte integrale del sistema tiene conto della somma degli errori nel tempo ed è in grado, ad esempio, di risolvere il problema dell'annullamento della corrente al raggiungimento della posizione richiesta. Nel caso di un motore elettrico, se raggiunge la posizione richiesta, senza la componente integrale la corrente va a zero, mentre con la componente integrale il sistema "ricorda" quanta fatica ha fatto

a raggiungerla e compenserà aumentando la corrente.

Per coniugare i tre sistemi nel PID in un microcontrollore è sufficiente:

- memorizzare l'errore corrente di posizione (parte P);
- memorizzare la differenza tra l'errore corrente e quello precedente (parte D);
- mantenere una somma degli errori man mano che procediamo (parte I);
- mettere i numeri risultanti in una formula che restituisce il valore di corrente da applicare al riscaldatore o motore per effettuare la correzione.

Nella pratica occorre considerare alcuni aspetti pratici e in particolare:

- bisogna (quasi per forza...) lavorare con numeri interi, perché a meno di non avere una CPU altamente performante la gestione dei numeri float rallenta l'esecuzione;
- siamo obbligati a dare un limite ai valori misurati, giacché non possiamo far fluire nel motore una corrente infinita;
- nella parte Integrale è inutile, anzi, dannoso, memorizzare una storia degli errori troppo "vecchia";
- la formula in sé è semplice, ma la determinazione dei suoi parametri no.

Semplificando, potremmo scrivere la formula in questo modo :

$$I = C_p \cdot p + C_d \cdot d + C_i \cdot i$$

dove C_p , C_d e C_i sono 3 costanti da determinare, p è l'errore sulla posizione, d è la sua variazione nel tempo, i è la "somma" degli errori nel tempo, mentre I è la corrente necessaria alla correzione.

La determinazione delle costanti e la gestione della parte integrale degli errori dipendono dalla dimensione dei numeri interi nella CPU e dalla tipologia del controllo desiderato.

Web Forum e supporto tecnico

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

www.elettronica.in.it/webforum

LED COB

LIGHTS

**100 LUMEN
WATT**



Realizza lettere, numeri per pannelli pubblicitari, illumina al meglio i prodotti nella tua vetrina e tanto altro con le barrette a LED COB. Ricoperte da gomma siliconica sono realizzate su supporto in alluminio per una migliore dissipazione di calore. Utilissime per svariati utilizzi!



Cod. LEDCOB200B



Cod. LEDCOB200W



Cod. LEDCOB200V



Cod. LEDCOB200R

BARRETTE DI LED COB 12V-15W LUNGHEZZA 20 CM

Tipo di LED: COB chip
Angolo del fascio: 120°
Flusso luminoso: 100 lumen/watt
Vita media: 50.000 ore
Alimentazione: 12 VDC
Potenza: 15W
Consumo: 900 mA
Temperatura di funzionamento: da -20°C a +60°C
Dimensioni: 200x10x3 mm
Peso: 11 grammi

€5,90

A BARRETTA

**Collega
insieme più
barrette di
colori
differenti!**

Prezzi IVA inclusa.

MAX 1
barretta di
LED COB

ALIMENTATORE SWITCHING
12V - 15W - 1,25A

€5,00

Cod. POS15W12V



MAX 2
barrette di
LED COB

ALIMENTATORE
SWITCHING
12V - 36W - 3A

€8,50

Cod. ADL-40-12



Alimentale con:

MAX 6
barrette di
LED COB

ALIMENTATORE
SWITCHING
12V - 100W -
8,5A

€20,00

Cod. MW10012



Vodafone sta lavorando alla realizzazione della prima Open RAN europea, che garantirà l'accesso a Internet a banda larga in Europa. Per la realizzazione dell'infrastruttura, l'operatore ha scelto di appoggiarsi ai server Dell EMC PowerEdge, sistemi rugged pensati per installazioni in aree caratterizzate da condizioni climatiche avverse, con temperature estreme. Vodafone inizierà aggiornando 2.500 delle sue stazioni radio in UK, per poi estendersi verso altri Paesi. RAN è l'acronimo di Radio Access Network e le relative reti rappresentano l'interfaccia fra gli apparati radio e la rete "core". Tipicamente i dispositivi RAN sono architetture "chiuse", nel senso che possono funzionare solo in abbinamento a specifiche apparecchiature.

» Radio Access Network Open



Per superare questo limite è stato creato, a cura di un apposito consorzio (formato da operatori di rete mobile, vendor e istituzioni accademiche e di ricerca che operano nel RAN) cui di recente ha aderito anche Google, lo standard Open RAN,

che serve ad applicare le caratteristiche della RAN ad hardware generici. Proprio secondo Google, RAN è il principale building block per ridurre il costo di proprietà, la scala e la complessità dei sistemi; per questo gli operatori di tutto il mondo hanno intrapreso il percorso per applicare i principi di disaggregazione, cloud e software centricity. Open RAN si traduce in vantaggi per operatori e community attraverso un ecosistema di vendor aperto e diversificato; in quest'ottica si colloca il connubio tra Vodafone e Dell Technologies. Le reti aperte e standardizzate saranno fonte di innovazione e di opportunità di business per il settore delle telecomunicazioni e per rendere possibili alti livelli di innovazione. cloud.google.com

» Motherboard e mini PC Asrock per l'IoT

ASRock Industrial Computer presenta la sua ultima scheda madre basata su processori Intel Atom x6000E, Intel Pentium e Celeron N e J (Elkhart Lake). Caratterizzata da prestazioni grafiche e CPU avanzate con funzionalità IoT integrate e supporto per la longevità, la nuova serie offre I/O avanzati incentrati sull'industria, tra cui doppie porte LAN Gigabit Ethernet, doppi display fino a 4K a 60Hz e doppi storage con SATA 3.0 e M.2 (Key M, 2242/2260/2280) per accelerare lo sviluppo dell'integrazione del sistema per la prossima generazione di applicazioni IoT in ambito domestico, aziendale, retail, bancario, alberghiero, manifatturiero, dei trasporti, sanitario ecc. Tra le novità, i mini PC NUC BOX-J6412, dotato di processore Intel Celeron J6412 e NUC BOX-J6426, basato su processore Intel Pentium J6426; tutti sono dotati di I/O e slot di espansione per collegare dispositivi industriali. La connettività ethernet è



dual Gigabit, mentre gli I/O sono due USB3.2 Gen2 (Type-C) due USB3.2 Gen2 (Type-A), uno USB2.0, un jack audio, oltre all'uscita HDMI 2.0b e DP 1.2 fino a 4096x2160@60Hz per un doppio display. Conta su due DDR4 SO-DIMM da 260 pin 3200 MHz fino a 32 GB di memoria e slot di espansione flessibili, inclusi doppi storage con un M.2 (Key M, 2242/2260/2280), uno SATA 3.0, più uno M.2 (Tasto E, 2230) per la connettività WiFi.

La serie iBOX 6000 senza ventola con modello iBOX-J6412, basata su processore Intel Celeron J6412 ha caratteristiche simili (32 GB di RAM SO-DIMM DDR4 3,2 GHz e interfacce video HDMI 2.0b and one DP 1.2, DP++ con risoluzione fino a 4096x2160@60Hz) ed è rivolto ad applicazioni embedded, industriali e di Edge Computing grazie alla sicurezza informatica e al supporto per la longevità. www.asrockind.com

› Sensori core ad apprendimento automatico

Qeexo, sviluppatore della piattaforma di apprendimento automatico automatico (ML) Qeexo AutoML ed STMicroelectronics, noto produttore di tecnologie per semiconduttori per tutte le applicazioni (mobilità intelligente, gestione efficiente dell'energia, controllo del movimento, implementazione su larga scala dell'Internet delle cose e del 5G) annunciano la disponibilità di i sensori core ad apprendimento automatico machine-learning core (MLC); questi sensori di movimento riducono il consumo energetico del sistema eseguendo localmente algoritmi relativi al rilevamento che altrimenti andrebbero eseguiti sul processore host, determinando un ulteriore impegno dello stesso e un incremento dei consumi

di elettricità. Utilizzando questi dati forniti dal sensore, Qeexo AutoML può generare automaticamente soluzioni di machine learning altamente ottimizzate per i dispositivi Edge, con latenza particolarmente bassa, consumo energetico contenuto e minimo ingombro di memoria.

Queste soluzioni consentono di superare i limiti computazionali del die e quelli delle dimensioni della memoria, con efficienti modelli di apprendimento automatico applicati ai sensori. Qeexo è la prima azienda ad automatizzare l'apprendimento automatico end-to-end per dispositivi embedded edge (classe Cortex M0-M4). La piattaforma Qeexo AutoML, con un semplice clic consente di utilizzare i dati dei sensori per creare ra-

› Dati IoT critici nelle strutture commerciali

Semtech Corporation ha annunciato la sua collaborazione con DIC Corp. per lo sviluppo della linea di sensori HattéTotte che comunicano tramite lo standard LoRaWAN e possono misurare la temperatura, l'umidità e la quantità di luce nell'ambiente. L'integrazione con LoRaWAN consente un basso consumo energetico e comunicazioni a lunga distanza, rendendo i sensori ideali per applicazioni in smart building, centri commerciali, spazi di lavoro, abitazioni ed altro ancora. Nello sviluppo dei sensori HattéTotte,

DIC Group ha puntato sulla semplicità e il design, ma anche sulla flessibilità di installazione, che non solo offre vantaggi economici ai gestori dell'edificio, ma che ha anche potenziali funzioni di sicurezza. La facilità di installazione dei sensori HattéTotte deriva dal design sottile, a strappo e a bastoncino del sensore, che consente di installare il sensore quasi ovunque in un edificio intelligente.

Una volta installato il sensore, l'implementazione è semplice con una facile configurazione tramite un per-

sonal computer in cui l'utente finale è in grado di monitorare la temperatura, l'umidità e la luce all'interno di uno spazio e automatizzare di conseguenza per il massimo comfort. I sensori HattéTotte sono sottili, disponibili in versione normale o impermeabile, facili da installare con la funzione peel and stick e con la batteria integrata possono fornire un'autonomia di 5 anni. L'utilizzo di LoRaWAN da parte di DIC per i suoi sensori HattéTotte semplifica l'implementazione delle soluzioni Internet of Things. La piattaforma da dispositivo (end device) a cloud LoRa di Semtech è una soluzione a lungo raggio e a bassa potenza adottata a livello globale per applicazioni IoT, che consente lo sviluppo e l'implementazione rapidi di reti IoT, gateway, sensori, moduli e prodotti a lungo raggio a bassissima potenza, convenienti e Servizi IoT in tutto il mondo. I dispositivi LoRa di Semtech forniscono il livello di comunicazione per lo standard LoRaWAN, che è mantenuto da LoRa Alliance, un'alleanza IoT aperta per applicazioni LPWAN (Low Power Wide Area Network) che è stata utilizzata per implementare reti IoT in oltre 100 paesi. Semtech è un membro fondatore della LoRa Alliance.

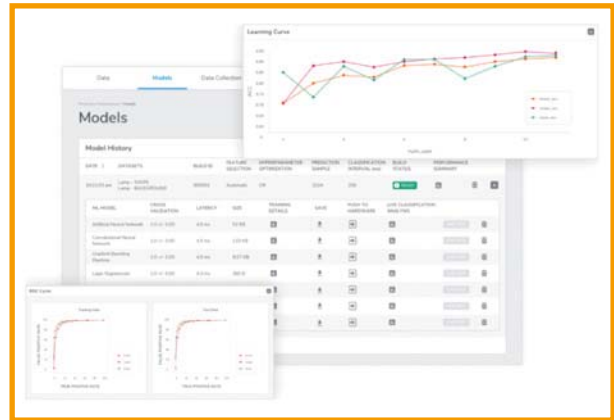
www.semtech.com



pidamente soluzioni di apprendimento automatico per ambienti altamente vincolati con applicazioni industriali, IoT, indossabili, automotive, mobile e altro ancora.

In tutto il mondo oltre 300 milioni di dispositivi sono dotati di intelligenza artificiale basata sul sistema Qeexo AutoML. Il lavoro congiunto di ST e Qeexo ha consentito agli sviluppatori di applicazioni di creare e distribuire rapidamente algoritmi di apprendimento automatico sui sensori MLC di ST senza consumare risorse. L'integrazione dell'MLC nei sensori ST, inclusi LSM6DSOX o ISM330DHCX, riduce i volumi di trasferimento dei dati del sistema e minimizza l'elaborazione di rete, oltre a limitare il consumo energetico del sistema di parecchi ordini di grandezza, fornendo al contempo un avanzato rilevamento degli eventi, logica di attivazione ed elaborazione edge in tempo reale.

www.st.com



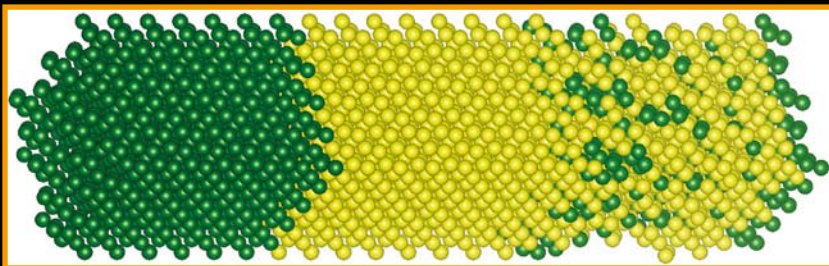
› I fisici addestrano l'AI a simulare ammassi atomici

I modelli di machine learning possono simulare bene le situazioni, ma spesso per l'addestramento richiedono molti dati, costosi da ottenere. Nel mondo della Fisica, un nuovo approccio mostra che i fisici possono prestare la loro esperienza agli algoritmi di apprendimento automatico, aiutandoli ad allenarsi su simulazioni costituite da pochi atomi, quindi prevedere il comportamento di sistemi con centinaia di atomi. In futuro, tecniche simili potrebbero persino caratterizzare microchip con miliardi di atomi, permettendo di prevedere guasti prima che si verifichino. I ricercatori hanno iniziato con unità simulate di 16 atomi di silicio e germanio e hanno impiegato computer ad alte prestazioni per calcolare le interazioni

quantomeccaniche tra gli elettroni degli atomi. Data una certa disposizione degli atomi, la simulazione ha generato caratteristiche a livello di unità come le sue bande di energia e i livelli di energia disponibili per i suoi elettroni. Ogni unità di simulazione è un insieme di forme 3D con pareti piatte, una per ogni atomo. Le pareti di una forma erano definite da punti equidistanti tra l'atomo e i suoi vicini (insieme, le forme si adattano perfettamente in quella che viene chiamata una tassellatura Voronoi). I loro set di allenamento consistevano in non più di 357 esempi. I ricercatori hanno addestrato due diversi tipi di modelli, una rete neurale e una foresta casuale, o un insieme di alberi decisionali, e li hanno

testati su tre diversi tipi di strutture, confrontando i loro dati con quelli di simulazioni dettagliate. Le prime strutture erano "superreticoli ideali", che potrebbero contenere diversi strati atomici di silicio puro, seguiti da diversi strati di germanio puro e così via. Le seconde erano "eterostrutture non ideali", in cui un dato strato poteva variare nel suo spessore o contenere difetti. In terzo luogo c'erano "eterostrutture fabbricate", che avevano sezioni di silicio puro e sezioni di leghe di silicio-germanio. I casi di test contenevano fino a 544 atomi. In tutte le condizioni, le previsioni delle foreste casuali differivano dai risultati della simulazione dal 3,7% al 19% e le reti neurali differivano dal 2,3% al 9,6%. La ricerca ha poi previsto un'operazione inversa: date le bande energetiche di un materiale, il loro sistema ne prevedeva la disposizione atomica: un tale sistema si assomiglia alla diagnosi dei guasti nei chip dei computer. Se manca la conducibilità di un semiconduttore potrebbe esserci difetto. La struttura simulata trova applicazione anche con altri tipi di materiali.

www.colorado.edu





Ecosistema MPLAB® di strumenti Cloud

Scopri, Configura e Sviluppa: un Ecosistema per Tutte le Tue Idee

L'ecosistema di strumenti cloud MPLAB è una soluzione online completa per utenti di qualsiasi livello di esperienza, per scoprire, configurare, sviluppare ed eseguire il debug di applicazioni embedded con microcontroller PIC® e AVR® (MCU).

- Accesso intuitivo allo sviluppo di MCU PIC e AVR con strumenti MPLAB integrati
- Prototipazione rapida con MCU PIC e AVR di Microchip utilizzando schede Curiosity
- Nessuna installazione di software per iniziare a sviluppare soluzioni MCU PIC e AVR

I progettisti che utilizzano MCU PIC e AVR possono visitare la pagina MPLAB Cloud Tools e iniziare a sviluppare in base alla loro fase di progettazione.

- Cerca e Scopri: accedi a MPLAB Discover per trovare progetti di codice sorgente pienamente configurati e completi
- Configura il codice: configura facilmente le applicazioni software con MPLAB Code Configuration
- Sviluppo e debug: lo sviluppo, il debug e la distribuzione di applicazioni di progetto direttamente dal browser preferito possono essere completati senza alcuna installazione software con MPLAB Xpres IDE



microchip.com/MPLABcloudtools

Il nome e il logo Microchip, il logo Microchip, AVR, MPLAB e PIC sono marchi registrati di Microchip Technology Incorporated negli U.S.A. e in altri Stati. Tutti gli altri marchi appartengono ai rispettivi titolari registrati.
© 2021 Microchip Technology Inc. Tutti i diritti riservati.
DS50003138A, MEC2381A-ITA-07-21

Driver backlight Maxim stabile in temperatura

Maxim Integrated presenta il driver per retroilluminazione a LED MAX25512, single chip a quattro canali e a bassa tensione con convertitore boost integrato; è l'unica soluzione integrata che mantiene la luminosità piena e costante dei display automotive anche in condizioni di avviamento a freddo estremo, fino a una tensione d'ingresso di 3V.

Gli odierni sistemi start-stop aumentano il risparmio di carburante, ma implicano instabilità dell'alimentazione tanto da rendere difficile, per il controller della retroilluminazione, mantenere lo stesso livello di luminosità del display durante il riavvio del motore. La retroilluminazione del display può essere influenzata dagli avviamenti a freddo, dove il picco di assorbimento scarica la batteria dell'auto abbastanza da far spegnere e riaccendere il display. Il driver MAX25512 funziona fino a 3V senza l'aggiunta di un convertitore pre-



boost, proteggendo il display da queste interruzioni dell'alimentazione. L'elevata efficienza del 91% con funzionamento a 2,2 MHz garantiscono una luminosità completa e costante dei display in ogni condizione. Funzioni diagnostiche come SHORT to GND su ciascun sink di

corrente permettono al microcontrollore di gestire il PWM del singolo canale attraverso l'interfaccia I²C. Inoltre una funzione di regolazione ibrida riduce le interferenze elettromagnetiche (EMI) e migliora il rapporto di regolazione.

www.maximintegrated.com

MCU 10 volte più veloci da TI



Le nuove MCU Sitara AM2x di Texas Instruments consentono di raggiungere una capacità di elaborazione 10 volte superiore rispetto alle MCU basate su Flash, candidandosi a competere con i processori odierni in applicazioni come l'automazione di fabbrica, la robotica, i sistemi automobilistici e la gestione sostenibile dell'energia. Costruito attorno ai core MCU Arm ad alte prestazioni, il portafoglio MCU Sitara AM2x include dispositivi single e multicore che funzionano a velocità fino a 1 GHz e integra periferiche e acceleratori specializzati. Strumenti accessibili

e software di facile utilizzo, semplificano la valutazione e riducono la complessità e i costi complessivi della progettazione. Gli MCU AM243x, la prima famiglia di dispositivi disponibili nel portafoglio AM2x, hanno fino a quattro core Arm Cortex-R5F, ciascuno in grado di funzionare fino a 800 MHz, consumando meno di 1W.

Con gli AM243x è possibile sfruttare stack di protocollo per supportare EtherNet/IP, EtherCAT, PROFINET, IO-Link Master, nonché applicare crittografia e meccanismi di sicurezza funzionale integrati. www.ti.com

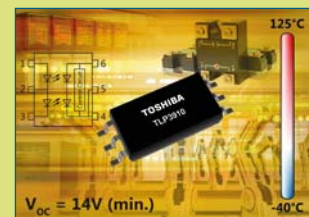
Fotoaccoppiatore HV per SSR da Toshiba

Toshiba Electronics presenta un nuovo fotoaccoppiatore in package SO6L da 3,84x10x2,1mm, adatto per pilotare i gate di MOSFET di potenza ad alta tensione realizzando relé allo stato solido (SSR) isolati galvanicamente. Gli SSR sono dispositivi relé a semiconduttore che incorporano un fototriac, un fototransistor e un fototiristore come dispositivo di uscita, a seconda che debbano commutare tensioni alternate o continue. Un fotoaccoppiatore come il TLP3910 può costituire la parte di accoppiamento ottico che pilota un MOSFET in qualità di elemento d'uscita.

Pilotare un MOSFET di potenza ad alta tensione con una tensione di gate di 10 V o superiore, attualmente richiede il collegamento serie degli stadi d'uscita di due optoisolatori, ma oggi il TLP3910 può farcela da solo, in virtù della tensione di riposo di 14V. Il circuito di scarica inte-

grato nel TLP3910 consente un tempo di spegnimento (toff) di 0,1 ms e un tempo di accensione (ton) di 0,3 ms. Il TLP3910 è il primo accoppiatore Toshiba con tensione di isolamento minima (BVS) di 5.000 Vrms pur mantenendo le prestazioni degli altri componenti Toshiba. Questo, insieme alla distanza di dispersione di 8 mm e alle approvazioni UL/cUL, consente l'utilizzo in apparecchiature industriali azionate a 400 Vca e altre applicazioni in cui la sicurezza elettrica è fondamentale. La temperatura di funzionamento è da -40 a +125 °C.

www.toshiba.com



U-blox porta la connettività WiF6 e Bluetooth 5 sulle schede M-2 NXP



U-blox presenta due schede di espansione WiFi 5, WiFi 6 e Bluetooth basate sui moduli JODY, per gli evaluation kit dei processori i.MX di NXP, con fattore di forma M.2. La scheda M2-JODY-W3 comprende un modulo u-blox JODY-W3 WiFi 6 e Bluetooth 5.1 ed è basata su chipset NXP 88Q9098 che supporta IEEE 802.11ax e

Bluetooth 5.1 dual mode, nonché WiFi dual band simultanee a 2,4 GHz e 5 GHz utilizzando due MAC e un'antenna MIMO 2x2 in ciascuna banda. Supporta il funzionamento simultaneo WiFi + Bluetooth e le modalità WiFi access-point, stazione o WiFi Direct. La M2-JODY-W2 è basata su modulo u-blox JODY-W2 WiFi 5 + Blue-

tooth 5, che utilizza il chipset NXP 88W8987 per supportare Bluetooth LE 5 dual-mode, WiFi dual-band standard 802.11a/b/g/n/ac, WiFi e Bluetooth simultaneo e la sua funzionalità è identica a quella del JODY-W3. Per entrambe le schede è disponibile un adattatore da microSD a M.2.

www.u-blox.com

Convertitori buck isolati STM per automotive e industria

ST presenta i converter CC/CC A6986I e L6986I, adatti ad applicazioni automobilistiche e industriali fino a 5 W.

La loro topologia buck isolata riduce i componenti esterni richiesti dai sistemi che richiedono una tensione primaria non isolata regolata e una o più tensioni secondarie isolate. La tensione secondaria è regolata dal rapporto di spire del trasformatore e non è necessario alcun fotoaccoppiatore per la regolazione.

Gli utilizzi comuni sono l'alimentazione di gate driver per MOSFET SiC e IGBT che richiedono rail di alimentazione isolati asimmetrici e l'alimentazione di interfacce isolate come RS-232, I²C e SPI; ma anche, per i convertitori A6986I qualificati AEC-Q100 Grade-1, i caricabatterie di bordo (OBC) per autoveicoli e le unità di trazione per veicoli ibridi/elettrici (H/EV). L'ampia gamma di tensioni di ingresso, da 4 a 38V soddisfano le specifiche di avviamento a freddo e scarico del carico per i sistemi automobilistici. L'L6986I di livello indu-



Single-chip isolated buck converters enhanced for flexibility and stability



striale è ideale per azionamenti motore, sistemi di condizionamento dell'energia solare, gruppi di continuità (UPS) e saldatrici. I convertitori funzionano in modalità PWM forzata con controllo in modalità corrente di picco per regolare la tensione di uscita primaria. Un limitatore di corrente negativa esteso sul lato primario offre ulteriore flessibilità per garantire una corrente secondaria adeguata utilizzando un tipico rapporto di bobina del trasformatore. I disposi-

tivi sono dotati di soft-start programmabile per migliorare l'affidabilità e il rilevamento della corrente impulso per impulso, per la protezione, a corrente costante nel lato primario. Un tempo di soppressione del rilevamento della corrente preimpostato di 300 ns filtra le oscillazioni generate dall'induttanza di dispersione del trasformatore e garantisce una maggiore stabilità durante le transizioni di commutazione high-side.

www.st.com/iso-buck-converters

Primato di Keysight e Qualcomm: 10 Gbps con la doppia connettività 5G

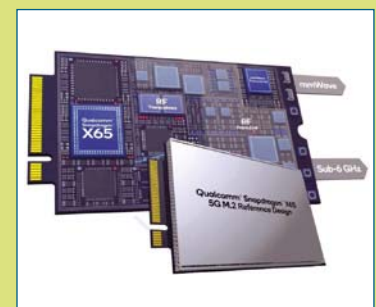
Keysight Technologies ha implementato una connessione dati a 10 Gbps utilizzando la doppia connettività 5G (NR-DC) in collaborazione con Qualcomm; allo scopo sono stati utilizzati la piattaforma di emulazione di rete Keysight e il sistema Modem-RF Snapdragon X65 5G con modulo antenna Qualcomm

QTM545 mmWave per aggregare lo spettro 5G mmWave e sub-6GHz. NR-DC consente ai dispositivi 5G NR di connettere e aggregare più celle NR, portando a velocità di trasferimento dati smagiori.

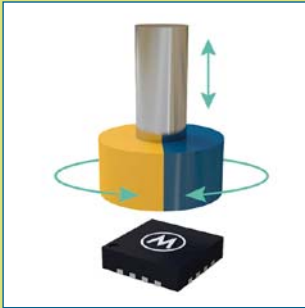
La dimostrazione ha utilizzato lo spettro sia nella gamma di frequenze 1 (FR1) che FR2 (mmWave), che consente alle

apparecchiature utente 5G (UE) di sfruttare ampie larghezze di banda e modulazione di alto ordine. Ciò consentirà agli operatori mobile di distribuire dinamicamente servizi 5G in aree congestionate dove la larghezza di banda deve essere ottimizzata per le applicazioni avanzate.

www.keysight.com



Simulatore magnetico di Melexis



Melexis ha introdotto uno strumento di simulazione online gratuito per la valutazione dei sensori magnetici, che permette di progettare e simulare i magneti basati sui sensori Triaxis 2D/3D di Melexis. Il simulatore copre applicazioni sia lineari che rotative, simula le prestazioni del sensore nell'applicazione con un alto livello di precisione, inclusa l'immunità ai campi vaganti. La simulazione con questo livello di precisione può migliorare e accelerare significativamente il ciclo di progettazione dei

moduli sensore a magneti. La simulazione di magneti multipolari non è semplice, perché implica una fisica complessa; ecco perché i simulatori del genere sono generalmente costosi. Melexis ha incluso quante più capacità di simulazione possibile ed offre lo strumento gratis. Lo strumento di simulazione online copre la progettazione meccanica e magnetica, per modalità di funzionamento sia lineare che rotativo, con o senza immunità ai campi vaganti inclusa. Lo strumento supporta anche la progettazione in tempo reale, fornendo risultati non appena un parametro viene modificato. Basandosi sul framework Streamlit e sulla libreria Python per formule analitiche per forme magnetiche semplici (magpylib), Melexis ha creato un catalogo di magneti utilizzabile per sviluppare e simulare le applicazioni. www.melexis.com

Piattaforma di sviluppo IoT da Renesas

Renesas Electronics propone una piattaforma di progettazione IoT che semplifica la prototipazione: si chiama Quick-Connect IoT ed è costituita da schede e interfacce standardizzate, che consentono ai progettisti di collegare rapidamente e facilmente un'ampia gamma di sensori alle schede di sviluppo MCU. Il sistema fornisce anche blocchi software di base portatili tra le schede. Renesas ha collaborato con Diligent alla creazione di una nuova interfaccia Pmod I²C-Bus ampliata, di tipo 6A e standardizzato il connettore Pmod 6A Diligent per nuovi sensori Pmod e kit di sviluppo MCU. Le schede MCU avranno due connettori Pmod, uno per un sensore Pmod e un altro per un modulo di comunicazione. www.renesas.com



Renesas ha anche ridefinito le API software comuni e il codice HAL per un'ampia varietà di sensori incorporati nell'ambiente di sviluppo integrato di Renesas e2 Studio: invece di dover scrivere e testare codice driver, i progettisti non devono fare altro che selezionare graficamente il loro sensore e scrivere poche righe di codice. Tutto lo sforzo di integrazione e configurazione viene svolto dietro le quinte, riducendo il tempo per arrivare a una soluzione funzionante. www.renesas.com

Sensore di velocità d'ingranaggio GMR da Allegro

Allegro MicroSystems ha introdotto un nuovo sensore di velocità GMR che misura la rotazione di ingranaggi ferromagnetici. L'integrato sensore è siglato ATS19480 e si applica alle trasmissioni di veicoli ibridi e puramente elettrici, nonché ad applicazioni industriali che richiedono solo informazioni sulla velocità di rotazione dei macchinari. Combinando la tecnologia GMR avanzata con i principali algoritmi di livello automobilistico e la tecnologia di imballaggio, Allegro offre ora un portafoglio completo di sensori di trasmissione all'avanguardia che soddisfa le esigenze degli sviluppatori e dei produttori di sistemi di oggi.

L'ATS19480 aggiunge un protocollo di sola velocità al circuito integrato del sensore di velocità e direzione di trasmissione ATS19580 recentemente rilasciato dall'azienda, stabilendo un nuovo standard per il rilevamento della velocità e consentendo agli sviluppatori di raggiungere i massimi livelli di capacità e adattabilità nel sistema. Con un traferro leader del settore che è più grande del 50% rispetto alle opzioni esistenti, migliora la flessibilità di progettazione, espande il margine di progettazione

e la capacità di tolleranza e facilita una gamma più ampia di posizioni di installazione dei sensori.

Come nostra più recente soluzione GMR a chip singolo per il rilevamento dei denti degli ingranaggi, l'ATS19480 ottimizza le prestazioni per le applicazioni emergenti dei veicoli elettrici.

L'integrazione monolitica consente al sensore ATS19480 di ottenere eccezionali prestazioni nel sistema e un rilevamento della velocità estremamente accurato. Il case SIP (Single Inline Package) a tre pin ospita il circuito integrato, il magnete e la protezione EMC. La compensazione avanzata del segnale elimina le condizioni flatline causate dalla dinamica del sistema e il rilevamento differenziale protegge dai campi di dispersione di modo comune.

L'ATS19480 include la diagnostica ASIL B integrata e un processo di progettazione di sicurezza certificato per la segnalazione di guasti opzionale e un funzionamento superiore in condizioni operative difficili può aiutare a ridurre i tassi di guasto, insieme ai resi dei clienti e alle richieste di assistenza in garanzia. www.allegromicro.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

10 NUMERI
A SOLI

€39

ANZICHÉ € 60,00

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

10
NUMERI
A SOLI

€45

ANZICHÉ € 60,00

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

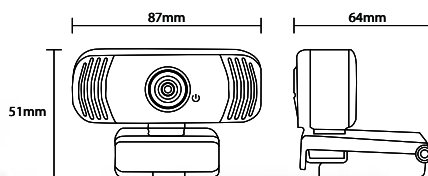
Prezzo IVA inclusa.



€24,00
Cod. FR795

WEBCAM FULL HD 1080P

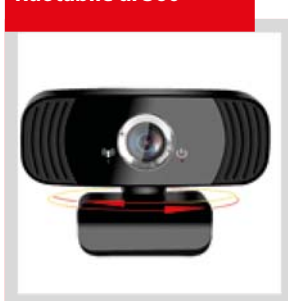
Webcam Full HD 1080P particolarmente indicata per seguire lezioni online, effettuare videoconferenze, videochat, videosorveglianza, registrazioni video, ecc. Dispone di lenti prive di distorsione, microfono omnidirezionale con riduzione del rumore, regolazione manuale della messa a fuoco, angolo di ripresa: 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



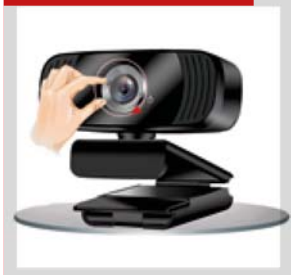
Ruotabile di 360°



Vari metodi di posizionamento



Supporta la messa a fuoco manuale



IN VETRINA

GENERATORE DI FUNZIONI JDS6600-60M



del Prof. MICHELE MENNITI

Generatore
di funzioni
basato sulla
tecnologia DDS,
indispensabile
per hobbisti
e riparatori.

N

el numero 247 vi abbiamo presentato un DSO con generatore di funzioni integrato della Hantek, un buon acquisto per chi deve dotarsi di entrambi gli strumenti, ma quando il nostro laboratorio è già attrezzato di Oscilloscopio, ciò che può servire è il solo generatore di funzioni. Oggi vi parleremo di uno strumento messo recentemente a listino da Futura Elettronica

col codice JT-JDS6600 e avente, come vedremo, prestazioni davvero interessanti. La storia di questo tipo di strumento è piuttosto recente, i primi apparecchi basati su un solo chip, in grado di generare le tre forme d'onda fondamentali: sinusoidale, quadra e triangolare, risalgono agli anni 80. I due circuiti integrati, in concorrenza tra loro erano l'XR2206 della Exar, in grado di generare frequenze tra 0,01 Hz ed 1 MHz e l'ICL8038 dell'Intersil, il cui range andava da 0,001 Hz a 300

kHz; entrambi si basavano su sistemi oscillatori richiedenti un gruppo RC esterno, per cui era necessario prevedere una coppia RC commutabile per ogni portata; inoltre erano dotati degli effetti di sweep, burst, variazione di duty cycle, elevata ampiezza del segnale che, in genere, con poca circuiteria esterna, prevedeva la compatibilità con le logiche digitali CMOS e TTL, che rappresentavano l'elettronica moderna dell'epoca. Chiaramente, come tutti gli oscillatori basati su componenti passivi come resistori e condensatori, questi strumenti peccavano di imprecisione ed instabilità, ma comunque la loro utilità era indubbia, soprattutto nei test di circuiti audio, in quanto coprivano l'intero range delle frequenze da 100 Hz a 20 kHz, e digitali, potendo svolgere la comoda funzione di base dei tempi, per testare circuiti divisori, flip-flop, contatori, trigger e molto altro.

I moderni generatori di funzione, con l'avvento dei microcontrollori e delle memorie eeprom, si sono arricchiti di molte funzionalità, con ulteriori forme d'onda, come rampe, impulsive, a gradino, semi-onde, ma soprattutto con la possibilità di generare le cosiddette forme d'onda arbitrarie, ossia segnali ad andamento ripetitivo ma non costante (come le classiche sinusoidi quadre e triangolari) realizzabili

dall'utente in base alle proprie esigenze, e quindi disegnabili punto per punto, con vantaggi inimmaginabili; alcuni di questi segnali si sono rivelati talmente utili nel tempo da essere integrati di default in moltissimi modelli di nuova generazione, per esempio il Sinc, l'ECG (elettrocardiogramma), il rumore di Gauss, le onde multitonali, ma il limite è dato dalla fantasia dell'utente. Tutti questi segnali, sia standard che arbitrari, sono ovviamente variabili in frequenza ed in ampiezza, con la possibilità di applicare ad essi un Offset DC, di variare la loro fase, opzione quest'ultima molto utile nei generatori a due canali.

In ultimo, ma non per questo meno importante, la possibilità di essere collegati ad un PC, inizialmente mediante porte seriali RS232, oggi mediante porte LAN o USB, permette il totale controllo via software di questi strumenti; infatti sono sempre dotati di un programma che emula il pannello comandi dello strumento e permette di impostarlo a piacimento via computer; inoltre comodissima risulta la possibilità di aggiornare il firmware del microcontrollore, per correggere eventuali bug iniziali e/o per implementare nuove funzionalità, o ancora per effettuare l'upgrade a modelli superiori. La tecnologia realizzativa dei generatori di forme d'onda professionali è definita "True Arbitrary", molto precisa e costosa, che permette di variare la frequenza di campionamento, cosa che garantisce la precisione del segnale generato uguale per qualsiasi frequenza, ma ciò crea complicazioni aumentando la velocità. I modelli meno sofisticati ormai si basano quasi tutti sulla tecnologia "DDS" (Direct Digital Synthesis), la cui limitazione, rispetto alla "True Arbitrary" è quella di operare a frequenza di campionamento fissa e ciò comporta il ricorso all'interpolazione per la generazione di un segnale, e quindi una maggiore imprecisione. In ogni caso comunque abbiamo a disposizione strumenti molto sofisticati, rispetto a quelli in uso nel passato.

IL GENERATORE JDS6600

Il generatore JDS6600 si basa appunto sulla tecnologia DDS e, nonostante il suo basso prezzo di listino, è un ottimo strumento, più che valido per il banco di hobbisti e riparatori.

Una volta aperta la confezione, troviamo lo strumento ed una eccellente dotazione di accessori: l'alimentatore esterno, due cavi BNC-coccodrillo rosso e nero, un cavo BNC maschio-maschio, un cavo USB per il collegamento al PC, il manuale in dotazione in inglese (**Fig. 1**).

La buona notizia è che l'autore del presente

Fig. 1
Il contenuto della confezione del JDS6600.





Fig. 2
Il generatore
JDS6600

articolo ha preparato un ottimo manuale, completamente in Italiano, decisamente più completo ed esaustivo di quello originale, che comprende anche le istruzioni per l'uso del software PC. Nella confezione troverete quindi le indicazioni per poter scaricare il relativo pdf.

Ci troviamo ora tra le mani un oggetto molto leggero, le caratteristiche costruttive sono ovviamente adeguate al prezzo pagato, infatti esso è racchiuso in un classico contenitore in plastica di formato standard, quelli che si trovano nei negozi di elettronica per le realizzazioni amatoriali, le cui dimensioni sono 194mm (L) x 178mm (P) x 69mm (H) (**Fig. 2**). La leggerezza di questo strumento è ovviamente dovuta al fatto che al suo interno ci sono solo la scheda principale dello strumento ed il pannello frontale che comprende il display TFT da 2,4", l'interruttore di accensione, i pulsanti e la manopola di regolazione (**Fig. 3**).

Tale leggerezza può al contempo essere ritenuta un difetto in quanto, in condizioni di eccessiva distanza dal circuito, una tensione dei cavi potrebbe causarne lo spostamento, quindi è bene fissarlo in qualche modo sul banco. D'altra parte, però, essa consente il facile trasporto ove occorra fare un intervento sul posto. In **Fig. 4** abbiamo riportato i pannelli anteriore e posteriore dello strumento, con la descrizione delle singole parti.

Il pannello anteriore, da sinistra a destra presenta: il tasto di accensione e spegnimento, il display grafico TFT, i tasti menu che servono per selezionare

le opzioni visualizzate di volta in volta sul bordo destro del display, i tasti funzione e la manopola generale che approfondiremo più avanti, i tre BNC, uno di ingresso esterno (per sincronismo o per misura della frequenza o periodo o conteggio di impulsi) e due di uscita (canale 1 e 2).

Il pannello posteriore ha un'interfaccia di estensione TTL, una porta di comunicazione USB ed la presa JACK per l'alimentatore esterno, che deve avere un'uscita di 5 V e 0,5 A, e spinotto con polo positivo centrale.

Fig. 3
L'interno del
generatore
JDS6600

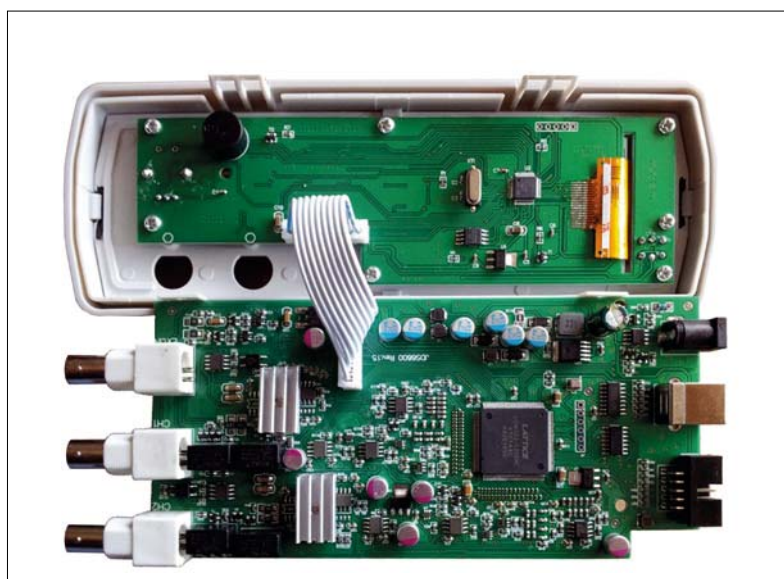
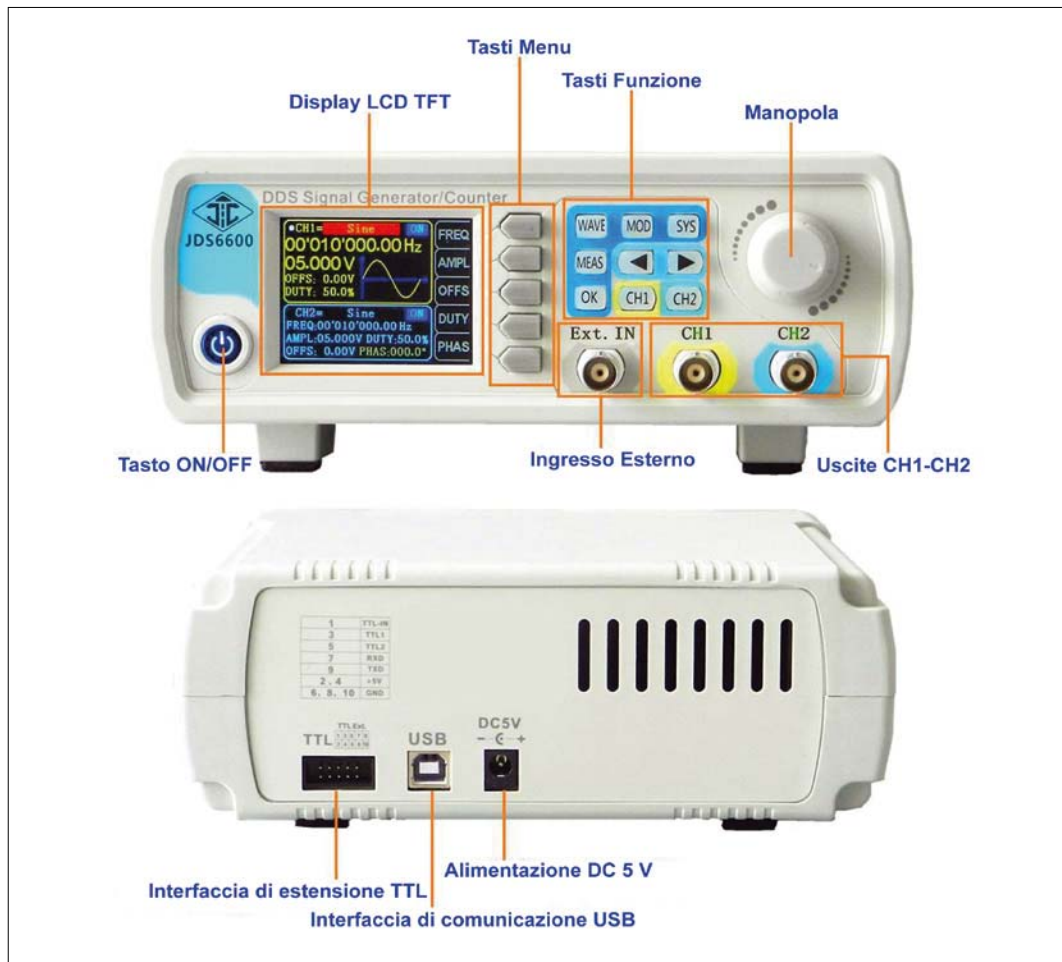


Fig. 4
I pannelli anteriore
e posteriore
del generatore
JDS6600.



La **Fig. 5** mostra l'organizzazione del display grafico del generatore. Il riquadro superiore più grande riporta di default i dati del CH1: forma d'onda, frequenza, ampiezza, offset, duty cycle,

stato dell'uscita e anteprima del segnale; nella parte inferiore sono riportate le stesse informazioni riguardanti il CH2, manca l'anteprima ma c'è l'indicazione della fase.

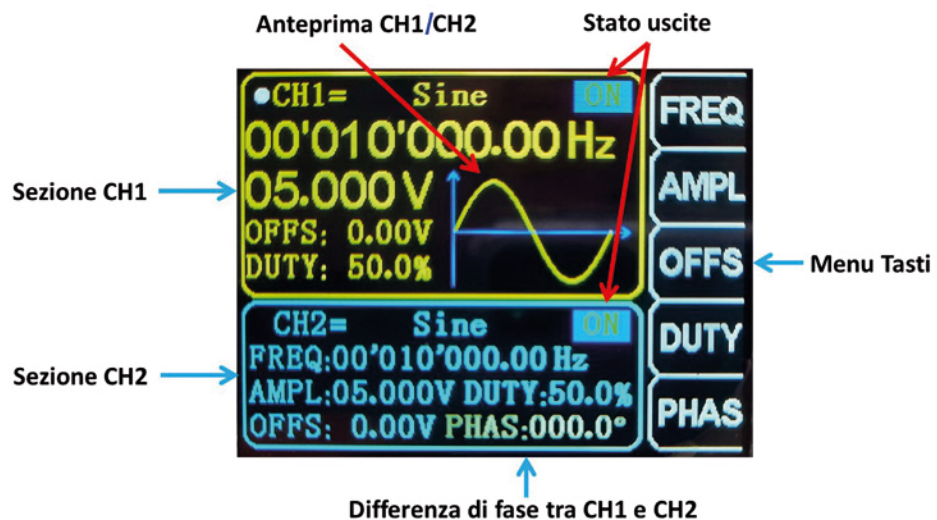


Fig. 5
Il display grafico.



Fig. 6
Impostazione della lingua.

In caso di necessità di avere sotto controllo il segnale del CH2 sarà sufficiente premere a lungo il relativo pulsante nel menu Funzioni e la visualizzazione dei due canali sarà invertita, il CH2 sarà portato nella parte superiore ed il CH1 in quello inferiore.

La posizione superiore stabilisce quale dei due canali è il master quando viene attivata la sincronizzazione, vedremo questa cosa in seguito.

LE CARATTERISTICHE TECNICHE

Il generatore di forme d'onda JDS6600 ha caratteristiche tecniche di tutto rispetto, riepiloghiamo

quelle principali.

Le frequenze massime delle forme d'onda principali sono: sinusoidale (60 MHz), quadra e triangolare (25 MHz), impulsiva (6 MHz); funzioni sweep e burst; l'ampiezza può variare tra 0 e 5-10-20 Vpp, in base al valore della frequenza; Offset DC -9.99 V ÷ 9.99 V; Fase tra 0 e 359,9°; 100 segnali e 60 onde arbitrarie memorizzabili.

Lo strumento inoltre è dotato delle ottime opzioni Frequenzimetro/Periodometro da 1 Hz a 100 MHz e Contatore fino a 4.294.967.295. L'elenco completo delle caratteristiche tecniche del generatore è riportato sul manuale in Italiano.

Tasto	Funzione
	Entra nel menu delle Forme d'Onda, da selezionare poi con i tasti funzione o con la manopola
	Entra nel menu Misura/Contatore da selezionare con la manopola, usare poi i tasti funzione per le opzioni
	Entra nel menu Modulazioni (Sweep CH1/CH2, Impulsi, Burst) da selezionare con la manopola, usare poi i tasti funzione per le opzioni
	Entra nel menu Sistema, usare manopola e tasti funzione per le opzioni
	Serve per attivare e disattivare contemporaneamente le uscite CH1/CH2.
	Tasti di spostamento all'interno del campo selezionato (p.es. Frequenza)
	Seleziona, attiva e disattiva l'uscita CH1, una lunga pressione del tasto imposta CH1 come canale principale sul display (sezione con l'anteprima).
	Seleziona, attiva e disattiva l'uscita CH2, una lunga pressione del tasto imposta CH2 come canale principale sul display (sezione con l'anteprima).

Fig. 7
Il menu dei tasti funzione.

Frequenza desiderata	Impostazione sul display portata MHz
5 MHz	05.000'000'00
1.530.852,63 MHz	01.530'852'63
160,333 kHz	00.160'333'00
50 Hz	00.000'050'00
0,1 Hz (100 mHz)	00.000'000'10

Unità di Misura	Simbolo	Scala	Conversione in Hz
MegaHertz	MHz	10 ⁶	1.000.000
KiloHertz	kHz	10 ³	1.000
Hertz	Hz	10 ⁰	1
milliHertz	mHz	10 ⁻³	0,001
microHertz	uHz	10 ⁻⁶	0.000.001

 **Fig. 8 (sopra)**
Impostazioni di frequenze.

 **Fig. 9 (a destra)**
La scala dell'Hertz.

USO DELLO STRUMENTO

Entriamo ora nel vivo della nostra recensione imparando ad usare lo strumento. Alla prima accensione occorre entrare nel menu di Sistema e impostare la lingua preferita (Inglese, Francese, Tedesco, Cinese), successivamente resterà memorizzata (**Fig. 6**).

Iniziamo ora col dare uno sguardo alla **Fig. 7**, una tabella riportante tutti i tasti funzione e le rispettive azioni, occorrerà tenerla costantemente sotto mano per i primi usi del generatore, fino a che non si raggiunge la necessaria padronanza.

La pressione del tasto **"OK"** attiva o disattiva contemporaneamente le uscite di CH1 e CH2, l'attivazione è mostrata dalla scritta ON sul display, nel rispettivo riquadro di ogni canale, e dall'accensione di un LED verde sopra il BNC di ogni canale. Invece la disattivazione fa apparire la scritta OFF e spegne i LED. Per attivare o disattivare separatamente i due canali si può ricorrere ai relativi tasti **"CH1"** e **"CH2"**; una prima pressione attiva le modifiche del canale, la seconda invece attiva o disattiva l'uscita del segnale sul relativo BNC. Per attribuire ad un canale il ruolo di "principale", assegnandogli così la sezione superiore del display e l'anteprima del segnale sul display stesso (di default è il CH1) occorre tenere premuto a lungo il tasto **"CH1"** o **"CH2"**.

Per impostare il tipo di forma d'onda occorre premere il tasto menu **"WAVE"**, quindi selezionare la forma d'onda desiderata mediante i tasti menu o la manopola, a quel punto sul menu del display appariranno i parametri impostabili per il segnale: Frequenza, Ampiezza, Offset, Duty Cycle e Fase. Sempre con i tasti menu si può selezionare ognuna di queste voci ed impostarne il valore. Per esempio selezionando la Frequenza (tasto **"FREQ"**) il display mostrerà il valore attuale con il cursore sul campo, usando le frecce   e la manopola si potrà impostare correttamente la frequenza di uscita del segnale, modificando il valore numerico. Dopo aver selezionato il campo della Frequenza, tenendo premuto il tasto **"FREQ"** si potrà modificare l'unità

di misura: uHz, mHz, Hz, kHz, MHz; durante la regolazione della frequenza occorre tener presente che il simbolo **"' "** rappresenta il separatore delle migliaia, mentre il **"."** è il punto decimale. Per esempio il range MHz si presenta come **"00.000'000'00"**: le prime due cifre da sinistra rappresentano i MHz, subito dopo c'è il separatore decimale, le successive tre cifre sono i kHz, a seguire le tre cifre degli Hz ed infine le due cifre dei decimi e centesimi di Hz. Per meglio comprendere questa cosa nella **Fig. 8** riportiamo alcuni esempi. Per azzerare tutte le cifre impostate occorre andare sulle prime due e ruotare la manopola verso sinistra. La tabella riportata in **Fig. 9** rappresenta la scala dell'Hertz.

Le forme d'onda disponibili di serie sono elencate nel manuale. Con i tasti **"AMPL"**, **"OFFS"**, **"DUTY"** e **"PHAS"** si potrà agire sugli altri parametri del segnale. I campi Offset, Duty Cycle e Phase, possono essere riportati al valore di default mantenendo premuto il relativo tasto menu.

La pressione del tasto **"MEAS"** apre il menu per le Misure esterne, quindi serve per misurare i segnali applicati al BNC **"Ext. IN"**, il cui range di ingresso va da 1 Hz a 100 MHz di frequenza e da 2 mVpp a 20 Vpp di ampiezza. Mediante il tasto menu **"FUNC"** si può scegliere tra la funzione **"Measure"** (Frequenzimetro) e quella **"Counter"** (contatore di impulsi), nel primo caso lo strumento misura la frequenza del segnale applicato al BNC, il periodo **"T"**, la larghezza dell'impulso positivo **"PW+"**, quella dell'impulso negativo **"PW-"**, il duty cycle **"Duty"**; inoltre è possibile regolare sia il **"GATE"** ossia la base dei tempi per la lettura, tra 0.01 s e 10 s, che la modalità (**"MODE"**), che può essere Periodometro (**"M.Period"**), per segnali fino a 2 kHz, con una risoluzione di 0.1 Hz, oppure Frequenzimetro (**"M.Freq"**) per segnali fino alla massima portata, con risoluzione di 1 Hz; in ogni caso la frequenza viene misurata in Hz. Nel secondo caso si attiva il contatore progressivo del numero di impulsi in entrata dal BNC, i due tasti **"▶"** e **"||"** avviano o mettono-

no in pausa il contatore mentre il tasto **"CLR"** lo azzerà. Con entrambi i tipi di misura il tasto **"COUP"** serve per impostare l'accoppiamento del segnale in DC o in AC.

La pressione del tasto **"MOD"** apre il menu delle Modulazioni di segnale, che possono essere selezionate mediante il tasto menu **"FUNC"**: Sweep CH1 o CH2 (**"S.F(CH1)"** o **"S.F(CH2)"**), Pulse (**"PLS(CH1)"**) e Burst (**"BST(CH1)"**), queste ultime due solo per il CH1.

La funzione **Sweep** serve per generare una sweep, cioè un segnale variabile in frequenza in funzione del tempo. Una volta selezionato il canale, è possibile impostare la frequenza del punto iniziale (**"Start"**) e quella del punto finale (**"End"**), all'interno dell'intervallo delle frequenze disponibili nel Generatore; la durata del tempo di scansione (**"Time"**) tra 0,1 s e 999,9 s, la direzione della scansione: dalla frequenza minima alla massima (**"Rise"**) o viceversa (**"Fall"**) oppure prima in una direzione e poi nell'altra (**"Rise&Fall"**); ancora il tipo di scansione può essere lineare (**"Linear"**) con cui il tempo viene diviso in parti uguali per ogni frequenza, oppure logaritmica (**"Logarithm"**) con cui si assegna un tempo maggiore alla parte iniziale del range e minore alle frequenze più alte. I due tasti **"ON"** e **"OFF"**, modificando il campo **"Control"**, impostano rispettivamente lo stato di esecuzione dello Sweep (**"Running"**) e quello di riposo (**"Stop"**), in questo caso il segnale assume la frequenza continua attiva in quel momento, all'interno del range impostato. Il campo in fondo alla schermata (**"Frequency"**) mostra costantemente la frequenza di scorrimento dello Sweep. Tutte le opzioni possono essere selezionate mediante le frecce  e modificate mediante il tasto **"CHG"** o con la manopola.

La Funzione **Pulse** genera dei singoli impulsi per i quali è possibile impostare la larghezza (**"PLS.W"**) e la durata (**"PLS.T"**), espressi entrambi in ns; questo sistema permette di ottenere dei segnali molto precisi, specialmente se si ha bisogno di un duty cycle avente un valore certo, invece di impostarlo in percentuale si lavora direttamente sulla durata della fase HIGH e quella dell'intero ciclo. In tal modo, per esempio, per ottenere un segnale con duty cycle perfetto al 50% è sufficiente impostare la durata della fase HIGH in base alla frequenza desiderata e quella dell'intero ciclo esattamente doppia della fase HIGH, in tal modo ci si assicura che la fase LOW sia assolutamente identica a quella HIGH; infatti se impostiamo la larghezza dell'impulso a 20 ns ed il tempo totale a 40 ns, la

durata della fase LOW sarà data da $40 - 20 = 20$ ns, quindi sarà identica alla fase HIGH. Gli altri parametri impostabili per questa funzione sono l'offset (**"PLS.Offset"**) da 0 a 100% (con 0 il segnale è equamente diviso tra V+ e V-, mentre con 100% la fase LOW coincide con 0 V) e l'ampiezza (**"Amplitude"**) da 0.01 V a 10 V. Infine i soliti tasti **"ON"** e **"OFF"** servono per attivare o disattivare la generazione degli impulsi (**"Control"**). Tutte le opzioni possono essere selezionate mediante le frecce  e modificate con la manopola.

La Funzione **Burst** genera dei treni di impulsi il cui numero può variare tra 1 e 1.048.575, ed ogni treno è separato dal successivo mediante una pausa costituita da una tensione di 0 V, la cui lunghezza è impostabile in frequenza. Il settaggio si effettua premendo il tasto **"NUM"** ed usando le frecce  e la manopola.

L'attivazione del treno avviene mediante il tasto **"ON"** ed il trigger (inizio del treno) può essere applicato secondo quattro modalità selezionabili mediante il tasto **"MODE"**:

- **Manuale ("Manual Trig.")**: il treno di impulsi si forma ogni volta che si preme il tasto **"ON"** oppure **"OK"**;

- **CH2 ("CH2 Trig.")**: lo strumento in questo caso effettua internamente il collegamento tra **"CH2"** e **"Ext. IN"**, quindi bisogna attivare la regolazione del CH2 e settare un segnale ad onda quadra o impulsiva con la frequenza alla quale si vuole far partire il treno di impulsi. Ipotizzando un treno di 10 impulsi con frequenza di 1 kHz, da inviare al circuito sotto test ogni 100ms, le operazioni da fare sono le seguenti:

- Premere **"WAVE"** e impostare **"CH1"** su **"Square"**; premere nuovamente **"WAVE"**, quindi **"FREQ"** ed impostare **"00'001.000'00 kHz"**, ampiezza 5 V, offset 0 V, duty cycle 50%;
- Premere **"CH2"** e quindi **"WAVE"** ed impostare **"Pulse"**, quindi premere di nuovo **"WAVE"**;
- settare il campo **"FREQ"** su **"00'000'100.00 mHz"**, ampiezza 5 V, duty cycle 50%, offset 0 V;
- Premere **"MOD"**, poi **"FUNC"** e selezionare **"BST(CH1)"**;
- Premere **"NUM"** e impostare **"0'000'010"**;
- Premere **"MODE"** e impostare **"CH2 Trig."**;
- Premere **"ON"** in modo da attivare il campo **"PControl"**.

- Esterno accoppiato in AC (**"Ext.Trig(AC)"**) in cui la sorgente di trigger deve essere collegata al BNC **"Ext. In"**;

- Esterno accoppiato in DC (**"Ext.Trig(DC)"**) in cui la sorgente di trigger deve essere collegata al



BNC "Ext. In". La pressione del tasto "SYS" apre il menu per le impostazioni di sistema ed i parametri dell'interfaccia utente; per l'uso di questo menu rimandiamo al manuale in Italiano.

IL SOFTWARE PER PC

Il Generatore di Funzioni JDS-6600 può essere pilotato da un PC, tramite l'interfaccia USB ed il software in dotazione, la cui ultima versione è scaricabile dal sito joy-it.net/en/products/JT-JDS6600, andando in basso nella sezione "download" e cliccando poi sulla voce "JDS6600 Software". Verrà scaricata una cartella compressa "JT-JDS6600.zip", nella quale, una volta decompressa, si troverà il file "autorun.exe", da eseguire per l'installazione del software. La prima maschera che si apre è il menu di installazione (Fig. 10). Occorre installare nell'ordine: il driver USB, il software VISA della National Instruments ed infine il software principale del Generatore di Funzioni. Il driver USB serve per il riconoscimento del chip CH340, che è l'interfaccia USB-seriale della scheda elettronica del Generatore, è sufficiente cliccare su "INSTALL". Per la procedura di installazione del software NI-Visa e del software principale rimandiamo al manuale in Italiano, così come per le manovre ed impostazioni preliminari d'uso. Una volta eseguito il software, cliccando sulla

scheda "Control Panel" si inizia a controllare lo strumento (Fig. 11). Il pannello di controllo è costituito da due sezioni identiche, quella di sinistra per il CH1 e l'altra per il CH2.

Nella parte centrale c'è il campo di regolazione della Fase tra i due segnali CH1 e CH2, in basso tutti gli switch per l'attivazione della sincronizzazione tra i due canali, ricordando che il master sarà sempre il CH1.

Descriviamo i campi di uno dei canali, dall'alto verso il basso:

- **Frequency:** impostazione numerica della frequenza del segnale, si può effettuare anche ruotando col mouse la "manopola di regolazione";
- **Unit:** unità di misura della frequenza del segnale;
- **Wave:** tipo di forma d'onda;
- **Preview:** anteprima del segnale impostato;
- **Amplitude:** ampiezza del segnale;
- **Offset:** offset DC del segnale;
- **X-Axys(dots):** numero di punti dell'anteprima orizzontale (2048, 4096, 8192);
- **Y-Axys(dots):** scala in V dell'anteprima verticale (0,04 ÷ 12V, Auto);
- **Duty:** duty cycle del segnale;
- **Adjust Items:** selezione della componente del segnale per la quale occorre impostare il valore di step (**Step Value**) nel campo sottostante, che sarà poi utilizzato nei rispettivi campi della maschera;

➔ Fig. 10
Installazione del
software per PC.



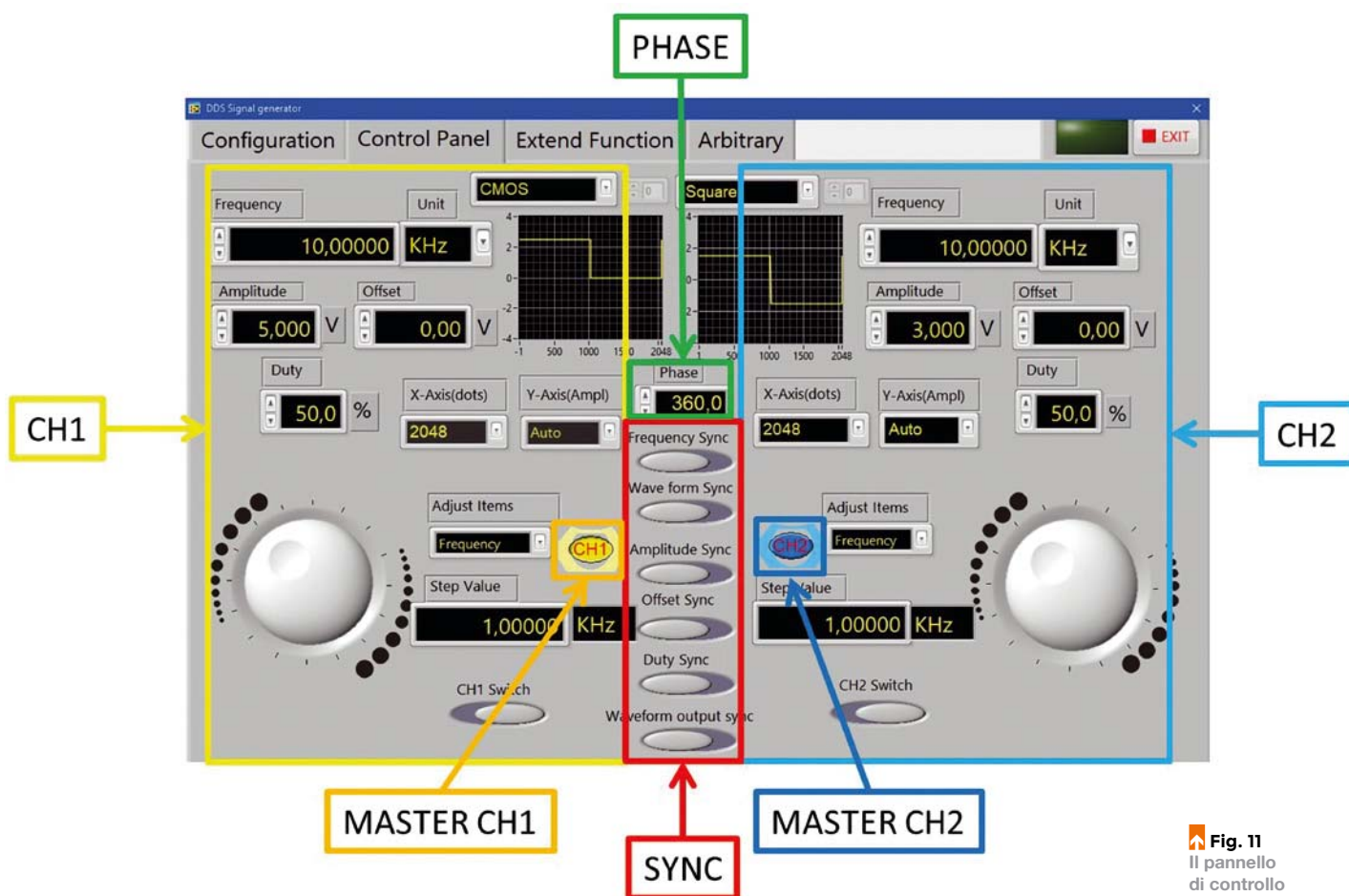


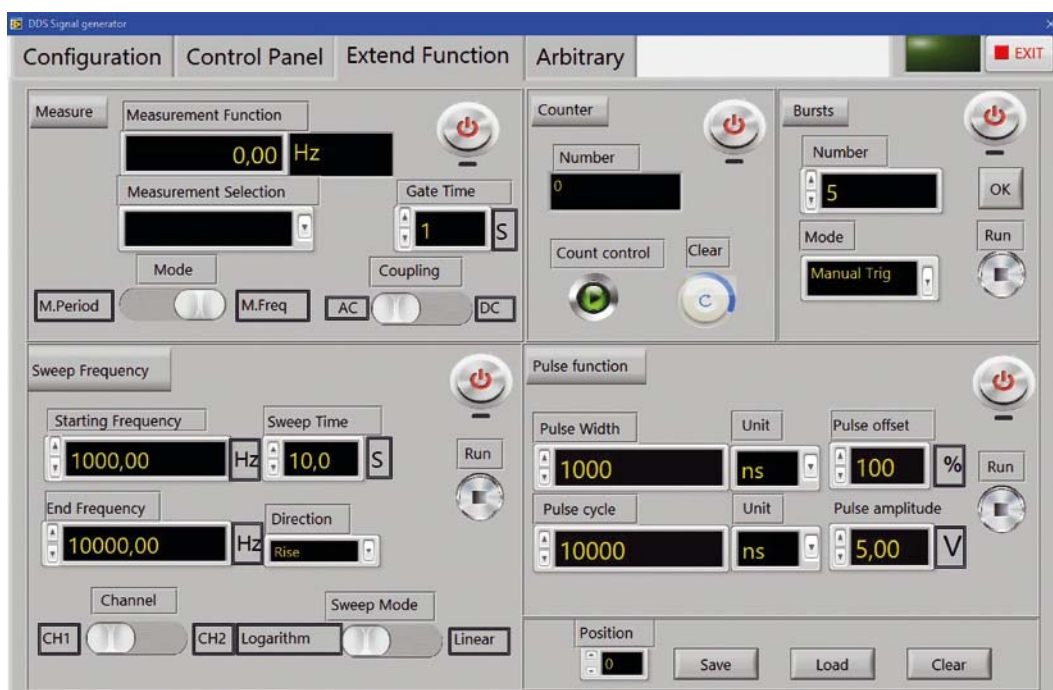
Fig. 11
Il pannello di controllo software.

- **CH1/2 Switch:** attiva o disattiva il relativo canale d'uscita;
- **MASTER CH1/CH2:** porta il relativo canale nella parte superiore del display del Generatore;
- **Sezione SYNC:** 6 selettori che spostati verso destra attivano la sincronizzazione del parametro tra CH1 e CH2. L'ultimo selettore in basso (**Waveform output sync**) attiva la sincronizzazione dei due CH1/2 Switch per cui il solo CH1 Switch accende o spegne entrambi i canali.

Selezionando la scheda "**Extend Function**" si accede alle funzioni speciali dello strumento (**Fig. 12**). Questa scheda è divisa in 6 sezioni, identificate da altrettanti riquadri, ognuno dei quali (eccetto l'ultimo) può essere attivato cliccando sul pulsante di "accensione" che da rosso diventa verde. Questi 5 pulsanti sono mutuamente esclusivi, in quanto ovviamente si può attivare una sola funzione per volta. Le funzioni speciali le abbiamo già descritte, qui ci limitiamo a dare alcune indicazioni sui campi. Le sezioni sono:

- **Measure:** lo switch "**Mode**" permette di scegliere tra la misura del periodo ("**M. Period**") o della frequenza ("**M. Freq**") del segnale. Un secondo switch permette di selezionare l'accoppiamento del segnale ("**Coupling**") in "**AC**" o in "**DC**". Nel campo "**Measurement Selection**" è possibile selezionare il valore da misurare: la frequenza "**Frequency**", il periodo "**Cycle**" (corrisponde a "**T**" sul display dello strumento), la larghezza dell'impulso positivo "**PW+**", quella dell'impulso negativo "**PW-**", il duty cycle "**Duty**"; il dato misurato sarà visibile nel campo superiore "**Measurement Function**", ed in quello accanto ad esso sarà visualizzata l'unità di misura, rispettivamente: "**Hz**", "**μs**" e "**%**". Infine è possibile impostare il "**GATE**" ossia la base dei tempi per la lettura, tra 0.01 s e 10 s.
- **Counter:** il conteggio degli impulsi in entrata dal BNC sarà mostrato nel campo "**Number**"; il pulsante verde "**Count control**" controlla l'avvio o la pausa del contatore, mentre il pulsante "**Clear**" lo azzerava. L'accoppiamento del segnale in DC o in AC si effettua con lo switch "**Coupling**" presente

Fig. 12
Le funzioni speciali.



nella sezione "Measure".

- **Burst:** il campo "Number" permette di impostare il numero di impulsi che compongono il treno, il campo "Mode" permette di selezionare la fonte del Trigger: manuale ("Manual Trig."), internamente dal CH2 ("CH2 Trig.") dello strumento o esterna, mediante un segnale applicato all'ingresso "Ext. In", che può provenire da una fonte di segnale esterna, accoppiata in AC ("Ext. Trig(AC)") o in DC ("Ext. Trig(DC)"). Nel caso dell'impostazione manuale il pulsante "OK" serve per far partire un treno ad ogni clic. Il pulsante "Run" serve per selezionare il treno oppure il segnale continuo.
- **Sweep Frequency:** nei due campi "Starting Frequency" e "End Frequency" si impostano rispettivamente la frequenza iniziale e quella finale dello sweep, esprimendole in Hz; nel campo "Sweep Time" si inserisce la durata totale dello sweep, in secondi; nel campo "Direction" occorre selezionare la direzione dello sweep tra "Rise", "Fall" e "Rise&Fall". Lo switch "Channel" seleziona il canale al quale applicare lo sweep, mentre lo switch "Sweep Mode" imposta l'andamento dello sweep tra logaritmico e lineare. Infine il pulsante "Run" serve per fermare lo sweep durante lo scorrimento, in tal caso il segnale assume la frequenza continua attiva in quel momento, all'interno del range impostato.
- **Pulse Function:** le impostazioni da inserire sono:

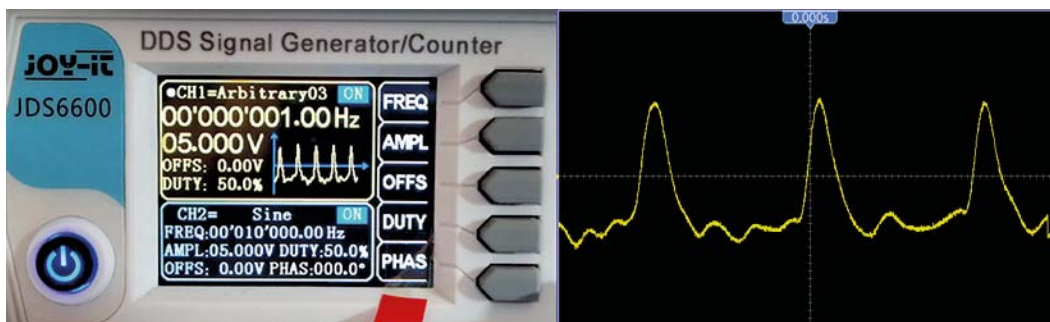
"Pulse Width" (larghezza dell'impulso, quindi la fase HIGH del segnale) in ns o μ s ("Unit"), "Pulse Cycle" (larghezza totale dell'intero impulso, quindi fase LOW + fase HIGH) sempre in ns o μ s ("Unit"), "Pulse Offset" (Offset DC pari al 50% dell'ampiezza del segnale) da 0 a 100%, "Pulse Amplitude" (ampiezza degli impulsi) da 0.01 V a 10 V. Il pulsante "Run" avvia o ferma la generazione degli impulsi.

- **Position:** questa sezione permette di memorizzare le impostazioni attuali dei due canali nella memoria del Generatore; l'unico campo variabile è quello della posizione di memoria (da 0 a 99); le impostazioni memorizzate nella posizione 0 saranno caricate di default ogni volta che si accende lo strumento.
Per effettuare una memorizzazione occorre seguire i seguenti passaggi:
 - Settare le impostazioni dei canali CH1 e CH2 con i comandi presenti sulla scheda "Control Panel" oppure direttamente sullo strumento;
 - Selezionare la posizione di memoria nel campo "Position";
 - Cliccare sul pulsante "Save".
 Per richiamare una impostazione memorizzata seguire invece i seguenti passaggi:
 - Selezionare la posizione di memoria nel campo "Position";
 - Cliccare su "Load".
 Le impostazioni presenti in una locazione di

memoria possono essere sovrascritte selezionandola e ripetendo l'operazione di "Save" oppure cancellate selezionandola e cliccando sul pulsante "Clear".

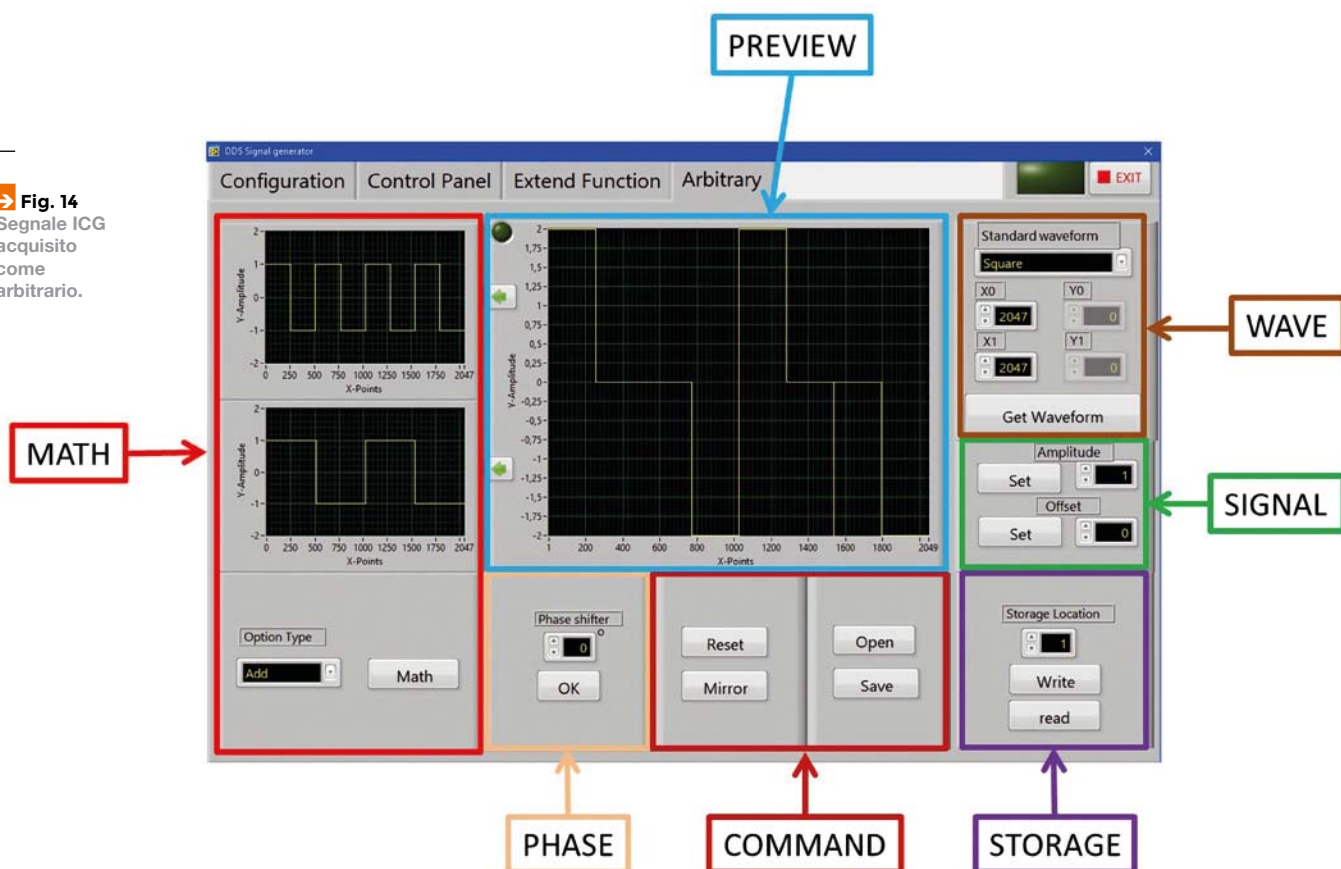
La scheda "Arbitrary" permette la gestione delle forme d'onde personalizzate. Vediamo le otto sezioni di questa scheda (Fig. 13)

- **PREVIEW:** è il display virtuale principale nel quale viene visualizzata la forma d'onda disegnata o creata con gli appositi comandi presenti nelle sezioni "Wave", "Signal", "Command" e "Phase". Successivamente la forma d'onda può essere trasferita e memorizzata in uno dei due display più piccoli della sezione "MATH" mediante le rispettive frecce verdi .
- **MATH:** è costituita da due riquadri nei quali è possibile visualizzare e memorizzare due diverse forme d'onda, mediante le rispettive frecce verdi, per poi eseguire una delle tre funzioni matematiche disponibili nel campo "Option Type": somma ("Add"), sottrazione ("Subtraction"), moltiplicazione ("Multiply"); cliccando sul pulsante "Math" il risultato sarà visualizzato nel display "PREVIEW".
- **WAVE:** questa sezione permette di selezionare, nel campo "Standard waveform" una delle forme d'onda standard previste dal Generatore, in questo caso saranno disponibili per le impostazioni solo i due campi "X0" e "X1", che permettono di stabilire la larghezza dell'onda intera, mediante il punto di partenza e quello di arrivo sull'asse orizzontale X. Se invece si seleziona la forma arbitraria in fondo all'elenco ("Customize polyline") si rendono disponibili anche i due campi "Y0" e "Y1" che operano sull'ampiezza dell'onda, in base alla scala presettata, impostando precisamente il punto di partenza e quello di arrivo sull'asse verticale Y. Il pulsante "Get Waveform" disegna istantaneamente sul display il segnale settato mediante i parametri X0/X1 ed eventualmente Y0/Y1. Nella modalità
- arbitraria è possibile anche disegnare il segnale a mano libera con il mouse, ma per ottenere una buona precisione è indispensabile l'uso di una tavoletta grafica in emulazione mouse.
- **SIGNAL:** in questa sezione è possibile settare sia l'ampiezza della scala del display PREVIEW che quella delle forme d'onda standard, mediante il campo "Amplitude" ed il pulsante "SET"; inoltre il campo "Offset" ed il pulsante "SET" permettono di impostare la quantità di OFFSET DC da applicare al segnale.
- **STORAGE:** in questa sezione è possibile selezionare la locazione di memoria ("Storage Location", da 0 a 60) nella quale salvare la forma d'onda creata, mediante il pulsante "Write", oppure dalla quale richiamare una forma d'onda salvata in precedenza, mediante il pulsante "Read". Da tener presente che questi segnali vengono memorizzati nel PC in cui è installato il software e non direttamente nel Generatore, che si limita a visualizzarli. Per memorizzare il segnale visualizzato in quel momento, anche nel Generatore, occorre agire direttamente nel menu "SYS" dello strumento, come già visto in precedenza.
- **COMMAND:** questa sezione è costituita da 4 pulsanti:
 - **Reset:** svuota il display PREVIEW per iniziare una nuova programmazione;
 - **Mirror:** inverte sull'asse Y il segnale visualizzato nel display PREVIEW;
 - **Save:** salva i 2048 valori Y della forma d'onda in un file di testo su supporto PC;
 - **Open:** apre un file di testo (tipo ms-dos) contenente i 2048 valori Y di una forma d'onda precedentemente salvata su un supporto PC. Questo comando può essere usato per importare valori di un segnale visualizzato e memorizzato da un oscilloscopio digitale (DSO), che poi sia in grado di esportarli sotto forma di file di testo o .csv (Excel, ma in questo caso va poi salvato in testo MS-DOS, dopo aver eliminato



◀ Fig. 13
Gestione delle onde arbitrarie

Fig. 14
Segnale ICG
acquisito
come
arbitrario.



dal file tutti i fogli vuoti); una volta visualizzato sul display PREVIEW, il segnale va salvato in una locazione di memoria con il comando **"Write"** e da quel momento in poi sarà sempre disponibile nel programma.

Rimandiamo al manuale in Italiano per un paio di esempi sulla creazione di una forma d'onda arbitraria e l'importazione di una forma d'onda visualizzata su un oscilloscopio. Qui ci limitiamo a mostrarvi il segnale importato da un DSO professionale collegato ad uno strumento medico cardiologico che rilevava dal polso di un paziente un segnale ICG (Impedence CardioGraphy, cardiografia ad impedenza), e memorizzato come segnale arbitrario nel generatore di funzioni in prova (**Fig. 14**). Nella parte sinistra dell'immagine si può notare che nel canale 1 è selezionata la forma d'onda arbitraria n. 3, impostata ad una frequenza di 1 Hz e ampiezza 5V, inoltre l'anteprima mostra la ricostruzione di tale segnale; nella parte destra vediamo la foto del DSO al quale abbiamo inviato tale segnale mediante il cavetto in dotazione BNC M-M. Ciò significa che da questo momento si dispone di

un segnale a ciclo continuo sul quale poter operare, per esempio rilevando misure, senza più bisogno di avere a disposizione il paziente e la strumentazione per acquisirlo.

CONCLUSIONI

Bene, arrivati a questo punto possiamo tranquillamente affermare di aver avuto tra le mani uno strumento che, pur non essendo catalogabile tra i professionali, vi si avvicina molto. Le sue caratteristiche e prestazioni sono di tutto rispetto e la sua presenza sul banco di un laboratorio di elettronica sarà di indubbia utilità. □



Cosa occorre?

Il generatore di segnali e frequenzimetro 2 CH - 60 MHz (cod. JT-JDS6600) è in vendita a Euro 169,00. Il prezzo si intende IVA compresa.

Il materiale va richiesto a:

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

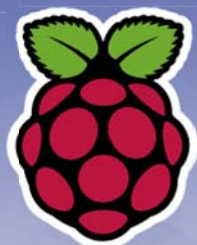
RASPBERRY PI 400!

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.

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.

MIRROR COPPO



di MAURIZIO ŠKERLIČ

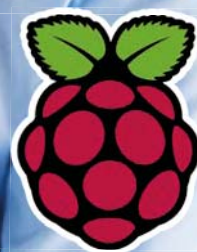
Specchio Smart basato su Amazon Alexa e Raspberry Pi, controllato da comandi vocali o gesti della mano può diventare l'interfaccia utente di una smart-home. Prima puntata.

Specchio, specchio delle mie brame... Chi non ricorda la fiaba dei fratelli Grimm dove Grimilde, la perfida matrigna di Biancaneve, interrogava il suo specchio magico su chi fosse la più bella del reame? Sarà un'idea insolita, però chi non desidererebbe possedere uno specchio magico?

Utilizzando una Raspberry Pi, un monitor LCD, uno specchio, l'assistente vocale Alexa e tanta pazienza, oggi il sogno può diventare realtà e possiamo realizzare uno specchio SMART con il quale realizzare una casa intelligente.

LA SCELTA DELLO SPECCHIO

Naturalmente non tutto è semplice perché non basta un normale specchio, ma ne serve uno semitrasparente (in inglese



viene chiamato One Way) come quello che viene utilizzato nelle stanze degli interrogatori e non permette di vedere chi c'è dall'altra parte. Difatti posizionando lo specchio davanti ad un monitor spento ci possiamo tranquillamente specchiare mentre se accendiamo il monitor vediamo l'immagine riprodotta dallo schermo in trasparenza. In commercio questi specchi esistono in vetro oppure in plexiglass. Trovare uno specchio One Way realizzato in vetro non è facile, quindi suggeriamo di acquistare una particolare pellicola specchiante adesiva da applicare sulla superficie di una lastra di vetro. Viene utilizzata per diminuire l'irradiazione della luce solare attraverso i vetri delle finestre, diminuendo il surriscaldamento delle stanze durante il periodo estivo.

Da prove effettuate abbiamo notato che se lo specchio, oppure la pellicola è molto riflettente, cioè ci si specchia bene, l'immagine riprodotta dal monitor si nota poco. Viceversa se lo specchio o la pellicola sono più trasparenti le immagini riprodotte dal monitor in trasparenza sono sì più nitide e ben visibili però a scapito dell'effetto specchiante. Per la realizzazione del nostro progetto abbiamo testato diversi tipi di pellicole prodotte dalla francese Reflectiv (<http://www.reflectiv.com>). Il miglior risultato come potere riflettente, cioè effetto specchio, lo abbiamo ottenuto con la pellicola One Way SOL101 che offre un buon compromesso fra trasparenza ed effetto specchiante. Questa pellicola,

della quale la **Tabella 1** riporta le caratteristiche, ha un potere riflettente del 75% (**Fig. 1**), valore accettabile se paragonato ad un comune specchio in vetro che ha un potere riflettente di circa 80%. Se invece volete un effetto specchio più marcato, cioè più simile a quello reale, potete scegliere un foglio con un potere riflettente maggiore tipo SOL102, però tenete presente che la trasparenza è inversamente proporzionale alla riflessione (la **Fig. 2** propone un confronto fra due pellicole). Come avete capito piccole variazioni di trasparenza incidono molto sulla luminosità dell'immagine trasmessa dal monitor, per questo non bisogna trascurare neanche lo spessore e la qualità del vetro. Forse non tutti sanno che il vetro comune non è del tutto trasparente. Se guardata di lato una comune lastra di vetro, vi accorgete che il colore del vetro tende al verde smeraldo. In commercio esiste un vetro ultra trasparente con basso tenore di ossido di ferro chiamato vetro museale usato nei musei per far risaltare l'opera, non altera i colori, la protegge dai raggi ultravioletti ed evita fastidiosi riflessi. In **Fig. 3** potete osservare la differenza tra una comune lastra di vetro e una di tipo ultra trasparente museale.

Anche lo spessore del vetro incide sulla trasparenza e soprattutto sul peso totale dello specchio. Un vetro da tre millimetri ha un peso specifico di 7,5 kg/m² mentre uno da quattro millimetri pesa 10 kg/m², quindi basta un solo millimetro in più

Tabella 1
Caratteristiche del
foglio riflettente
SOL101.

MATERIALE	Poliestere 36 micron
ADESIVO	Polimero acrilico 19 gr/m ²
TINTA	Argento metallizzato
PELLICOLA DI PROTEZIONE AMOVIBILE	Poliestere siliconato 23 micron
APPLICAZIONE	Interna
RIFLESSIONE SOLARE TOTALE	55 %
ASSORBIMENTO SOLARE TOTALE	33 %
TRASMISSIONE SOLARE TOTALE	12 %
TRASMISSIONE DELLA LUCE VISIBILE	16%
RIFLESSIONE DELLA LUCE VISIBILE	60%
ENERGIA SOLARE TOTALE RESPINTA	75 %
TRASMISSIONE RAGGI UV	5%
RESISTENZA TERMICA	da -20°C a 80°C
TEMPERATURA DI APPLICAZIONE	Minimo 5°C
DIMENSIONI	60" x 8,34'

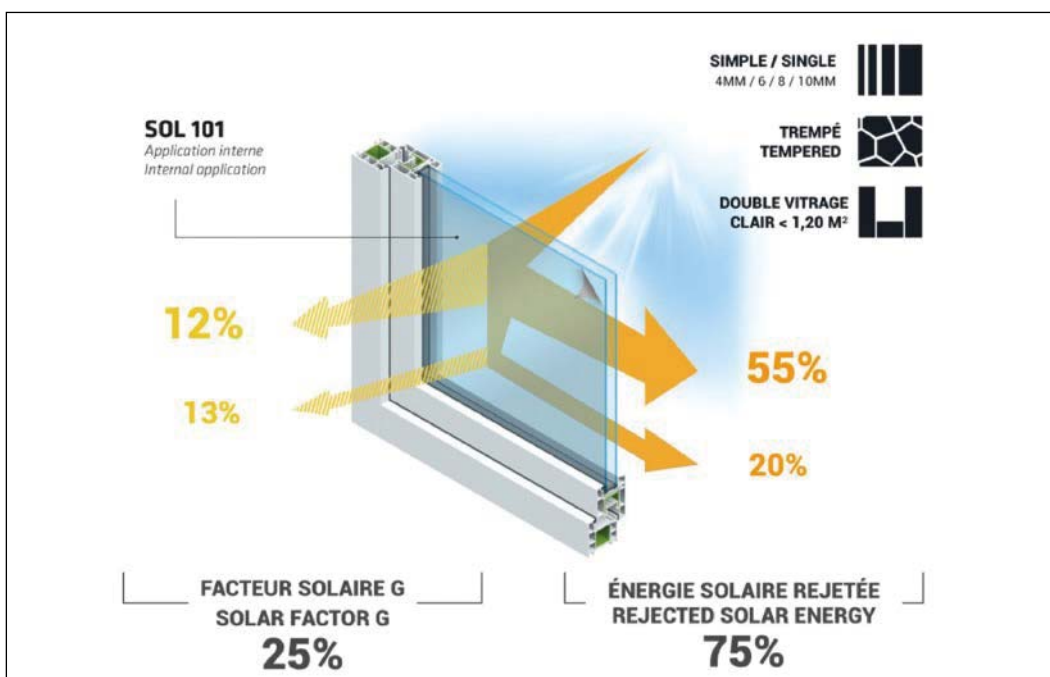


Fig. 1
Rappresentazione grafica delle caratteristiche del foglio riflettente SOL101.

di spessore per far aumentare il peso in modo considerevole.

Facciamo un esempio. Se volete realizzare uno specchio utilizzando un monitor da 21 pollici, le cui dimensioni sono circa 54 cm per 30 cm, la superficie sarà di 0,162 m². Utilizzando un vetro da tre millimetri il peso sarà di 1,2 kg mentre con un vetro da quattro millimetri il peso raggiungerà 1,6 kg. Nel caso invece vogliate strafare e realizzare uno specchio più grande utilizzando un monitor da 42 pollici, la cui superficie è di 0,5 m², probabilmente un vetro da tre millimetri risulterà troppo sottile con il rischio di frantumarsi. Dovrete per forza optare per un vetro più spesso e resistente. Se utilizzerete un vetro da quattro millimetri il peso sarà di 5 kg e aumenterà ulteriormente di un chilo se utilizzerete un vetro da cinque millimetri. Tenete presente in fase di progettazione che al peso del vetro va sommato anche il peso della cornice, del monitor, degli alimentatori e di tutti i circuiti elettronici.

Una volta scelto il tipo di vetro dovete applicare la pellicola riflettente adesiva. Prima bisogna sgrassare accuratamente la superficie del vetro con uno straccio imbevuto di solvente (alcool) e successivamente lavarlo accuratamente con acqua e sapone per eliminare eventuali aloni. Asciugate bene la superficie ed eliminate la polvere con un getto di aria compressa. Per esperienza prima di iniziare il lavoro vi consigliamo vivamente di aspirare bene la polvere dalla stanza e soprattutto dal piano di lavoro, ricordando di indossare guanti in lattice per

non lasciare impronte digitali tra vetro e pellicola adesiva. Procuratevi un vaporizzatore riempito con acqua distillata o demineralizzata, per non lasciare aloni di calcare, e bagnate abbondantemente la superficie del vetro. Per facilitare il posizionamento

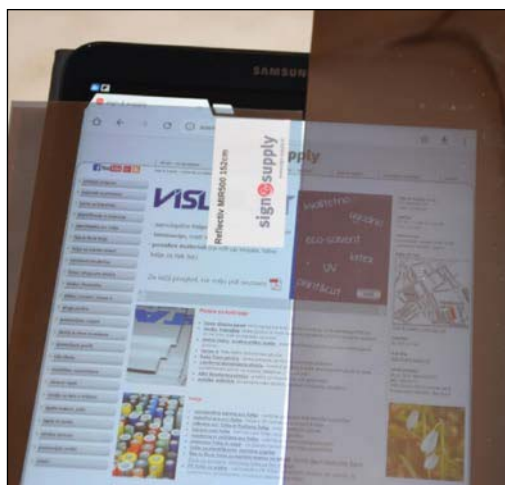


Fig. 2
Differenza di trasparenza delle pellicole One Way messe a confronto.



Fig. 3
Lastra di vetro comune (sotto) e lastra museale ultra trasparente (sopra).

della pellicola all'acqua distillata potete aggiungere alcune gocce (max. 6 gocce per litro) di detergente per stoviglie incolore. A chi interessa in commercio esistono liquidi specifici per l'applicazione di pellicole adesive su vetro. Fatevi aiutare da qualcuno e incollate la pellicola tenendo la lastra di vetro in posizione verticale in modo che la polvere non si depositi sulla superficie del vetro. Prima di staccare il foglio di protezione dalla pellicola consigliamo di vaporizzare il foglio con abbondante acqua per lavare via qualsiasi particella di polvere altrimenti l'elettricità statica generata dalla separazione dei due fogli attrarrà tutta la polvere sul lato adesivo e una volta applicato il foglio sul vetro si noteranno tutte le imperfezioni. Per separare la pellicola dal foglio protettivo consigliamo di attaccare sull'angolo un pezzetto di nastro adesivo su entrambe le superfici e poi lentamente separare i due fogli. Ricordatevi di attaccare il foglio sulla lastra di vetro stendendolo dall'alto verso il basso, eliminando le bolle di aria aiutandovi con un particolare raschietto di plastica flessibile rivestito di tessuto per evitare di non graffiare la delicata superficie riflettente.

Dopo aver lasciato riposare la pellicola per un paio di giorni, se si riformano eventuali bollicine di aria le stendete delicatamente con il raschietto e se necessario riscaldando la pellicola con un asciugacapelli. Se qualche granello di polvere è rimasto intrappolato sotto la pellicola non cercate di staccare il foglio adesivo dal vetro perché lo rovinereste irrimediabilmente.

LA SCELTA DEL MONITOR

Superata questa iniziale difficoltà, trattandosi di uno specchio SMART, in grado di fornire informazioni a schermo, avrete bisogno di un monitor. Noi consigliamo l'utilizzo di un monitor da 24 pollici con risoluzione Full HD a 1080p, che abbia un profilo stretto altrimenti, una volta realizzato il progetto, risulterà troppo spesso e la cornice sposterà troppo dal muro assomigliando più ad un pensile che uno specchio.

Tra le altre caratteristiche scegliete quello provvisto di prese d'ingresso e di uscita laterali e non posteriori, altrimenti il connettore HDMI sposterà nella parte posteriore aumentando lo spessore. Poiché la carcassa in plastica sarà rimossa, scegliete un monitor che sia facile da disassemblare. Tenete presente che i pannelli dei monitor non sono tutti uguali, le principali famiglie di pannelli a cristalli liquidi di tipo TFT presenti oggi in commercio sono prevalentemente di tre tipi TN (twisted

nematic), VA (vertical alignment) e IPS (in-plane switching).

I pannelli TN sono quelli con concezione cronologica più vecchia, hanno bassi costi di produzione e tempi di risposta bassissimi, con alti refresh rate. In poche parole sono molto veloci e particolarmente adatti e compatibili con tutte le tecnologie da gioco sviluppate negli ultimi anni. D'altra parte però i pannelli TN presentano i colori mediamente spenti e gli angoli di visione poco ampi, rendendo la visione laterale dei colori distorta e poco definita. Per eliminare questi difetti è stato concepito il pannello IPS che per la prima volta ha dotato un monitor di un'accuratezza di colore superiore, angoli di visualizzazione più elevati e una resa visiva migliore. Su questi pannelli non si notano sfasamento dei colori e perdita di nitidezza delle ombre, nemmeno guardandoli con un angolo elevato sia in verticale che in orizzontale. La riproduzione dei colori è ottima, e sono i prescelti da coloro che hanno bisogno di una fedeltà cromatica maggiore.

Anche se la tecnologia IPS costituisce il non plus ultra grazie alla loro riproduzione dei colori, agli angoli di visione elevati ed alla alta reattività, possiedono purtroppo un grande difetto, che avrete sicuramente notato anche voi, il basso contrasto. La profondità dei neri non è riprodotta così bene come su altri tipi di pannelli, in poche parole il nero non è nero ma una specie di grigio scuro. Poiché oggi la maggior parte dei pannelli commercializzati possiede un pannello di tipo IPS, sicuramente anche il vostro ne possiede uno. Provate a visualizzare sul vostro monitor un'immagine nera, magari spegnendo la luce della stanza, e capirete.

Questo difetto viene corretto dalla tecnologia VA la quale non rappresenta il nero tenendo accesa la retro illuminazione del pannello, ma lo fa spegnendo del tutto. Questo rende i neri realmente neri, eliminando quella tonalità grigiastra presente negli altri pannelli. Il contrasto statico è più elevato, possiede elevati angoli di visualizzazione e una ottima riproduzione del colore, tutte caratteristiche che li rendono i prediletti per la visione dei film, in cui i neri generalmente abbondano. L'ottima riproduzione del nero e soprattutto l'alto contrasto sono le caratteristiche principali che ci hanno convinto ad utilizzare questa tecnologia nel nostro progetto con risultati eccellenti. Un pannello IPS a causa della sua tecnologia, illumina anche le zone prive di immagini che dovrebbero essere buie, rovinando l'effetto specchio rendendo il tutto poco realistico. Fidatevi del nostro consiglio e utilizzate un pannello con tecnologia VA la differenza è abissale ed



inoltre paradossalmente un monitor con pannello VA è più economico di uno con tecnologia IPS. Per chi interessa nel nostro progetto abbiamo utilizzato un monitor della Benq modello GW2470, comunque qualsiasi altro monitor di qualsiasi marca che possiede tale tecnologia va benissimo. Lo specchio per essere "magico" deve saper rispondere alle nostre domande e richieste. Quindi è stato dotato dell'assistente vocale Alexa installato su una Raspberry Pi 3 B+.

Prima di spiegare come installare l'assistente vocale spiegheremo brevemente, a quelli che sono alle prime armi, come installare il sistema operativo Raspbian sulla microSD della Raspberry Pi.

IL SISTEMA OPERATIVO RASPBIAN

Procuratevi una micro SD con una capacità di memoria di almeno 8GB noi consigliamo una da 16GB, disponibile a catalogo Futura Elettronica con il codice "SDHC16GB", o meglio ancora da 32GB, a catalogo Futura Elettronica con il codice "SDHC32GB", non tanto per la sua capacità ma bensì per la maggiore velocità di lettura e scrittura. Se prendiamo in considerazione due schede della stessa capacità ad esempio 16GB contrassegnate con lo stesso simbolo di velocità ad esempio C10 UHS-I non possiedono mai la stessa velocità, peggio ancora alcuni modelli con capacità di memoria maggiore sono più lenti di quelli con capacità di memoria minore. Scegliere una microSD per ottenere il massimo delle prestazioni dalla nostra Raspberry Pi è importante. In rete si trovano innumerevoli strumenti per stabilire la reale velocità delle memorie SD, uno tra i tanti che vi possiamo consigliare è il programma CrystalDiskMark scaricabile gratuitamente dal sito ufficiale. Scoprirete con stupore che alcune SD dei brand più blasonati sono in realtà più lente di quelli meno conosciute. Bisogna far presente che la velocità dipende molto dal tipo di hardware usato durante il test e dalle applicazioni che inevitabilmente sono aperte in background. Test eseguiti su pc diversi daranno risultati diversi.

Scelta la microSD adeguata anche se nuova, per ogni evenienze, formattatela con il programma "SD Card Formatter", avviandolo come amministratore. Questo programma vi servirà anche in futuro per formattare le unità di memoria SD se non riconosciute correttamente dal sistema operativo e ripristinare la capacità piena di memoria della scheda.

Una volta che la formattazione della microSD è andata a buon fine, potete caricare su di essa il

sistema operativo Raspbian scaricabile dal sito ufficiale della Raspberry Pi.

Da dicembre 2020 la Raspberry Pi foundation ha deciso di ribattezzare Raspbian, il sistema operativo Debian per RPI, in "Raspberry Pi OS" e verrà distribuito in due versioni sia a 32 che 64bit. Per ora la versione a 64 bit è solo in versione beta installabile solo sulla RPI 3 e 4, per questo, finché non esce la release definitiva, consigliamo di installare la versione a 32 bit.

A titolo informativo l'immagine da noi installata è la versione "Debian buster". Scegliete l'immagine completa quella contenente il desktop e i programmi raccomandati, riportiamo il link diretto https://downloads.raspberrypi.org/raspbian_full_armhf/images/raspbian_full_armhf-2020-12-04/2020-12-02-raspbian-buster-armhf-full.zip. Dopo aver decompresso l'immagine, per caricarla sulla micro SD utilizzate il programma "Win32 Disk Imager". A seconda della velocità della scheda SD, la scrittura durerà parecchi minuti. Una volta terminata la scrittura, inserite la scheda nella Raspberry Pi, collegate l'uscita HDMI al monitor, collegate un mouse e una tastiera alla presa USB, collegate con un cavo ethernet la presa RJ45 alla rete LAN domestica e date tensione. La fase di boot durerà alcuni secondi e successivamente sul monitor apparirà il desktop di "X-windows". Lanciate "LXTerminal" ed eseguite il comando "sudo raspi-config". Si avvierà il programma di configurazione, dal menu "advanced option" scegliete il sottomenu "expand filesystem". Dal menu "advanced option" scegliete il sottomenu "resolution" e selezionate la risoluzione 1920 x 1080 50 Hz 16:9. Dal menu "interfacing option" selezionate il sottomenu "SSH" e abilitatelo. Fate lo stesso per il menu "I2C" e il menu "VNC", quest'ultimo vi servirà per controllare la Raspberry Pi da remoto con il programma "Real VNC viewer". Uscite dal programma e a richiesta salvate la configurazione.

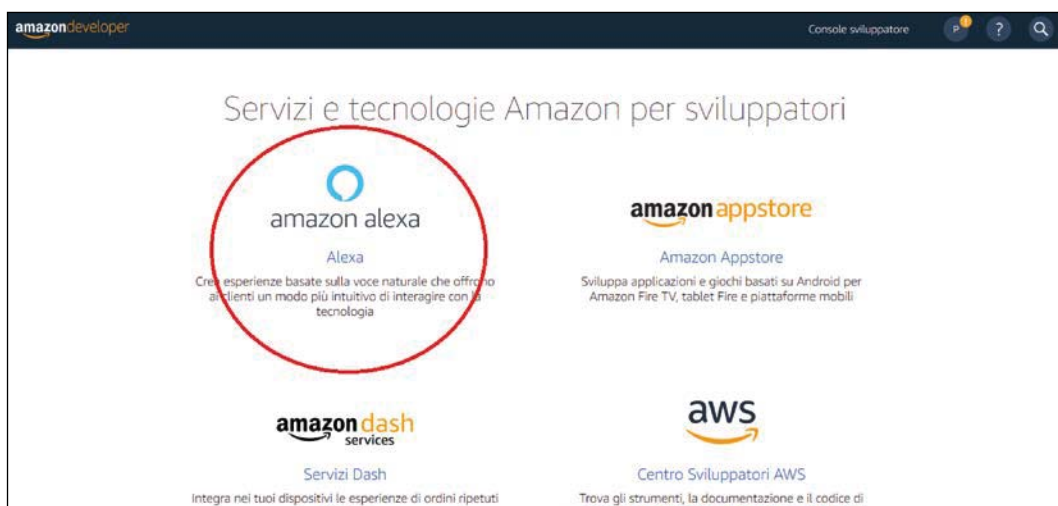
Dalla riga di comando eseguite il comando "cd /home/pi", successivamente lanciate il comando "sudo apt-get update". Una volta terminato eseguite il comando "sudo apt-get upgrade -y". Quest'ultimo passaggio può durare anche parecchi minuti abbiate pazienza.

CREAZIONE DELL'ACCOUNT AMAZON DEVELOPER SERVICE

Per installare l'assistente vocale Amazon Alexa vi servirà un account "Amazon developer service". Se non lo avete dovete crearlo visitando l'indirizzo <https://developer.amazon.com/it/>, selezionate la



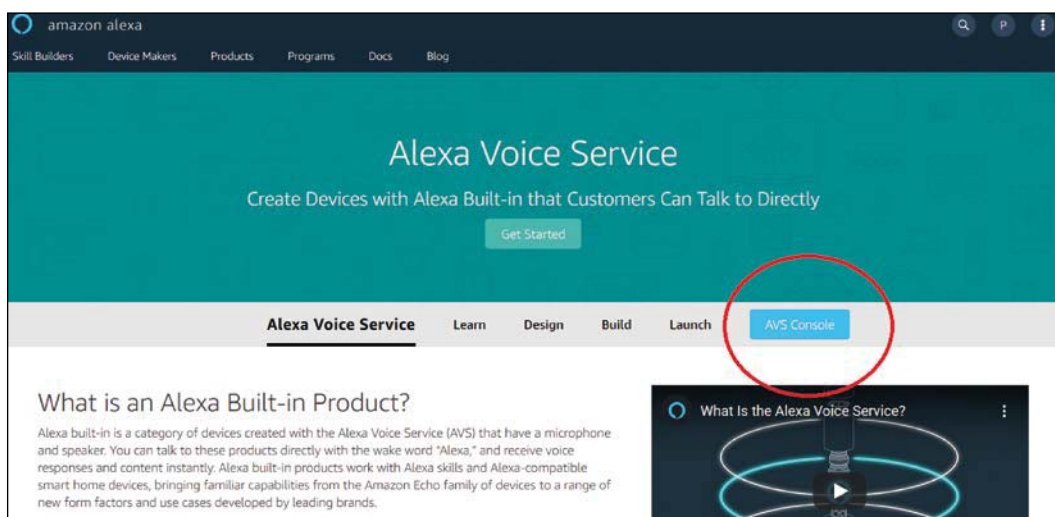
Fig. 4
Selezionate
"Amazon Alexa".



sezione "Amazon Alexa", come visibile in **Fig. 4**, e dal menu a tendina "Prodotti" selezionate "Amazon voice service". Si aprirà una nuova schermata dove selezionerete il pulsante "AVS console", come raffigurato in **Fig. 5**. Vi verrà chiesto di specificare l'indirizzo e-mail e la password del vostro account Amazon. Se non avete un account Amazon cliccate sul pulsante "Crea il tuo account Amazon" e seguite le istruzioni a video. Una volta terminata la registrazione vi ritroverete nella schermata di "Amazon developer service" dove dovrete cliccare sul pulsante in basso "Get started", e dal menu a tendina "Products", della nuova schermata, selezionate "Add new product". Compilate il form rispondendo ad alcune domande come visibile in **Fig. 6** e **Fig. 7**. In "Product name" inserite "mirrorcoppo", in "ProductID" digitate "mirrorcoppo", spuntate "Device with Alexa built-in", alla domanda "Will your device use a compa-

nion app?" spuntate "No", per "Product category" selezionate "other" e nella casella successiva inserite "prototype", nel riquadro "Brief product description" digitate "prototype", alla domanda "How will end users interact with your product?" selezionate "Hand free", e al resto delle domande rispondete spuntando la casella "No". Cliccate sul pulsante "Next". Nella pagina successiva selezionate "Create new profile", nei riquadri "Security Profile Name" inserite "mirrorcoppo" e come "Security Profile Description" digitate nuovamente "mirrorcoppo". Cliccate sul pulsante "Next". Come visibile in **Fig. 8** nella casella "Allowed origins" inserite gli URL "https://localhost:3000" e "http://localhost:3000" mentre in "Allowed return URLs" inserite gli URL "https://localhost:3000/authresponse" e "http://localhost:3000/authresponse". A questo punto per "Platform information" selezionate "Other devices

Fig. 5
Cliccate sul
pulsante "AVS
Console".



Product information
Tell us about your Product

Product name *

Product ID *

Please select your product type *

- ☐ Application with Alexa built-in
A standalone app. This includes apps on the web, Android, Kindle, iOS, FireTV, AppleTV, etc.
- ☒ Device with Alexa built-in
Physical product with the potential to have buttons, knobs, a touch screen, etc. Examples are speakers, headsets, televisions, set top boxes, appliances, etc. Includes Alexa Mobile Accessories.
- ☐ Alexa Gadget
An accessory that connects to a compatible Echo device via Bluetooth. For example, an animatronic that moves a motor in response to sound, or lights up if a notification arrives. [Learn More](#)

Will your device use a companion app? *

- ☐ Yes
- ☒ No

Product category *

Other (please specify)

Brief product description *

[How will end users interact with your product?](#)

[FEEDBACK](#)

Fig. 6
Il form da compilare.

and platforms" e nella casella "Client ID name" digitate "mirrorcoppo" e cliccate il pulsante "GENERATE ID". Copiate e conservate il "Client ID" e cliccate il pulsante "download".

Il file "config.json" precedentemente scaricato e il "Client ID" vi serviranno successivamente per l'attivazione di Alexa una volta installata sulla Raspberry Pi. Per terminare dovrete accettare i termini

[How will end users interact with your product?](#)

☐ Touch-initiated
A user's primary way to interact with Alexa is by tapping or holding a button.

☒ Hands-free
Hands-free products allow users to interact with Alexa by using their voice at a close distance.

☐ Far-field
Far-field products allow users to interact with Alexa by using their voice from a long distance.

Upload an image
This image is shown to end users on the [Manage your Content and Devices](#) page.

UPLOAD IMAGE
MAX: 142 X 130 PX

Do you intend to distribute this product commercially? *

- ☐ Yes
- ☒ No

Will your device be used for Alexa for Business? *

- ☐ Yes
- ☒ No

Is this device associated with one or more AWS IoT Core Accounts? *

- ☐ Yes
- ☒ No

Is this a children's product or is it otherwise directed to children younger than 13 years old? * [Learn More](#)

- ☐ Yes
- ☒ No

NEXT

[FEEDBACK](#)

Fig. 7
Compilate il form seguendo le istruzioni dell'articolo.

Fig. 8
Compilate il form
seguendo
le istruzioni.

Allowed origins ⓘ

https://www.example.com Add

http://localhost:3000 ×

https://localhost:3000 ×

Allowed return URLs ⓘ

https://www.example.com/login.php Add

http://localhost:3000/authresponse ×

https://localhost:3000/authresponse ×

☒ I agree to the Amazon Developer Services Agreement, including the Alexa Voice Service Program Requirements.

FINISH

FEEDBACK

Italian (italiano) © 2010-2019, Amazon.com, Inc., o sue affiliate. Tutti i diritti riservati. Termini Documenti Forum Blog Alexa Developer Home

di servizio spuntando la casella in basso e cliccare sul tasto "Finish".

INSTALLAZIONE DI AMAZON ALEXA

Finalmente siamo pronti per l'installazione dell'assistente vocale Amazon Alexa. Accendete la Raspberry Pi e dal terminale, oppure da remoto con il programma *Putty*, vi spostate nella directory home "`cd /home/pi`" dove lancerete in successione i seguenti comandi:

```
"sudo wget https://raw.githubusercontent.com/alexav/avs-device-sdk/master/tools/Install/setup.sh \"  
\"sudo wget https://raw.githubusercontent.com/alexav/avs-device-sdk/master/tools/Install/genConfig.sh \"  
\"sudo wget https://raw.githubusercontent.com/alexav/avs-device-sdk/master/tools/Install/pi.sh"
```

Una volta terminato il download dei file dal repository di GitHub copiate nella cartella "`/home/pi`" il file "`config.json`", precedentemente scaricato da Amazon developer service, e lanciate il comando "`sudo bash setup.sh config.json [-s 123456]`". Apparirà una schermata dove vi chiederà di accettare i termini di licenza delle librerie di terze parti, confermate digitando con lettere maiuscole la parola "AGREE". Ad un certo punto il programma di installazione vi chiederà di leggere e accettare i termini e le condizioni di licenza della libreria "*Sensory*" utilizzata per il WWE (wake word engine), voi confermate digitando "yes". *Sensory TrulyHandsfree* è la libreria che riconosce la nostra voce quando pronunciamo il nome dell'assistente vocale "Alexa". Questa libreria è a pagamento e viene concesso l'utilizzo gratuito

per un periodo di prova di 120 giorni, trascorsi i quali "Amazon Alexa" non riconoscerà più la nostra voce e non risponderà alle chiamate vocali. Non preoccupatevi, con pochi comandi possiamo aggiornare e ricompilare la libreria. Per quelli che non desiderano ripetere la procedura tre volte l'anno dovranno contattare la ditta *Sensory* all'indirizzo URL "`https://www.sensory.com/support/contact/`" e acquistare la licenza. Dopo aver accettato i termini di licenza l'installazione proseguirà e durerà parecchio tempo, anche più di 30 minuti. Non allarmatevi se i led sulla Raspberry Pi non lampeggeranno oppure le scritte a terminale non scorreranno più, aspettate con pazienza. Per aggiornare la licenza di *Sensory* ogni quattro mesi dovete aggiornare le librerie scaricandole dal sito web github. Spostatevi nella cartella di installazione del programma *Sensory* con il comando "`cd ~/third-party/alexarpi`" e lanciate i seguenti comandi e accettate i termini del contratto di licenza:

```
sudo git reset --hard  
sudo git pull  
sudo ./bin/license.sh;
```

In casi estremi nel caso il "wake word engine" non funzionasse ancora bisognerà aggiornare anche l'assistente vocale Amazon Alexa ripetendo il procedimento di caricamento di "*avs-device-sdk*" da GitHub e ricompilarlo come spiegato sopra.

LA SCELTA DEL DISPOSITIVO AUDIO

La Raspberry Pi è provvista di un'uscita audio jack da 3,5 mm però non è provvista di un'entrata



microfono. Per sopperire a questa manchevolezza progettuale si possono adottare diverse soluzioni che noi abbiamo provato per voi e vi suggeriamo di utilizzare. La più economica è quella di usare un microfono USB a catalogo Futura Elettronica (codice USBMICMINI, visibile in **Fig. 9**), le prestazioni non saranno ottime poiché il microfono è integrato nella chiavetta USB e non può essere posizionato o indirizzato verso la fonte sonora. Una soluzione migliore si otterrà utilizzando il microfono USB con cavo, a catalogo Futura Elettronica (codice MICUSB, visibile in **Fig. 10**), oppure in alternativa potete utilizzare una sound card USB, visibile in **Fig. 11**. Le prestazioni in questo caso saranno migliori perché potete utilizzare l'uscita audio della scheda audio della chiavetta USB e non quella della Raspberry Pi la quale è risaputo presenta evidenti distorsioni. Prestazioni ancora migliori possono essere ottenute usando una scheda dedicata come la Respeaker della Seeed studio. Nel nostro prototipo abbiamo usato la scheda "ReSpeaker 4-Mic Linear Array Kit", vedi **Fig. 12**, provvista di ben 4 microfoni e di una circuiteria capace di filtrare ed elaborare il suono eliminando l'effetto eco. Siccome questa scheda non è proprio economica abbiamo pensato opportuno descrivere la realizzazione di una versione più economica usando la chiavetta audio USB ma pur sempre con prestazioni accettabili. L'uscita audio predefinita della Raspberry Pi è indirizzata verso la porta HDMI. Se volete prelevare il suono dalla presa jack, dovete lanciare da terminale il comando `"sudo raspi-config"` e scegliere dal menu `"advanced config"` il sottomenu audio. Qui scegliete forzare l'uscita attraverso la presa jack per le cuffie da 3,5 mm. Per utilizzare la scheda audio su chiavetta USB dovete accedere alla scrivania di *X-windows*, collegando la Raspberry Pi ad un monitor o al televisore con un cavo HDMI, oppure più semplicemente connettendovi da remoto con `"Real VNC viewer"`.

Nella parte alta a destra della scrivania cliccate con il tasto destro del mouse sull'icona dell'altoparlante di colore azzurro e selezionate il dispositivo `"USB audio device"`, sia per `"audio outputs"` sia per `"audio inputs"`. Dal menu `"input device settings"` impostate il controllo del volume del microfono al massimo e passaggio importantissimo non selezionate la funzione `"auto gain control"`, altrimenti avrete grossi problemi con l'attivazione dell'assistente vocale con la voce. Dal terminale lanciate il comando `"sudo nano /usr/share/alsa/alsa.conf"`, scorrete verso il basso e modificate `"defaults.ctl.card 0"` e `"defaults.pcm.card 0"` in `"defaults.ctl.card 1"`

e `"defaults.pcm.card 1"`.

Salvate e riavviate la Raspberry Pi.

Il miglior metodo per verificare il corretto funzionamento del microfono e dell'impostazione dell'uscita audio è quello di usare il programma grafico `"Audacity"`. Questo programma non è presente nell'immagine Raspbian e quindi bisogna installarlo manualmente lanciando da terminale il comando `"sudo apt-get install audacity"`.

Una volta installato potete lanciarlo dal menu `"Audio e video"` e una volta entrati nel programma impostate sia per il microfono sia per l'uscita audio il dispositivo `"USB audio device"`, mentre per verificare il corretto funzionamento del microfono,



Fig. 9
Microfono
USB mini.



Fig. 10
Microfono
USB con cavo.



Fig. 11
Adattatore
audio USB.

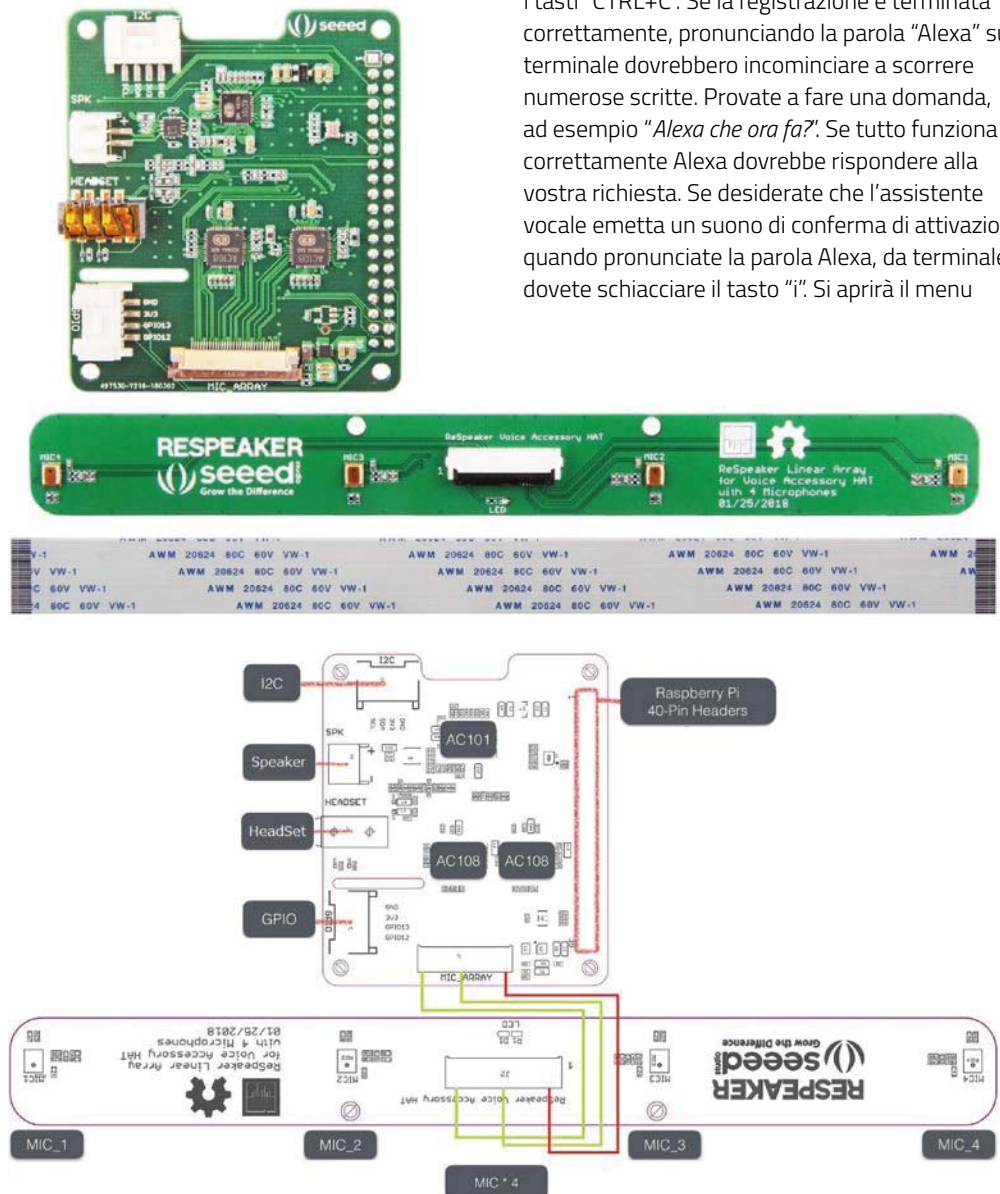
claccate sul VU meter del microfono per attivarlo, come visibile in **Fig. 13**. Provando a parlare da diverse posizioni nella stanza impostate l'amplificazione del microfono in modo tale che il segnale audio non vada in clipping.

ATTIVAZIONE DI AMAZON ALEXA

Ora è giunto il momento più interessante e tanto agognato, cioè quello di installare e attivare Amazon Alexa. Con il browser visitate la pagina web di Amazon all'indirizzo "amazon.com/us/code" e vi registrate con le credenziali di sviluppatore Amazon. Vi verrà chiesto di inserire il codice di autenticazio-

ne, generato dal programma che dovrete lanciare da terminale con i seguenti comandi `"cd /home/pi/"` seguito da `"sudo bash startsample.sh"`. Una volta eseguito il comando sul terminale incominceranno a scorrere velocemente verso l'alto numerose scritte, aguzzando la vista e aiutandovi con la barra di scorrimento laterale cercate il codice di autenticazione. Ricopiatelo nella finestra di registrazione di Amazon come visibile in **Fig. 14**. L'autorizzazione può durare anche 30 secondi prima di ricevere una risposta di conferma. In questa fase il programma che gira a terminale non deve essere fermato fino al termine della registrazione. Quindi, per poter leggere il codice sopra menzionato, non vi venga in mente di arrestare il programma utilizzando i tasti "CTRL+C". Se la registrazione è terminata correttamente, pronunciando la parola "Alexa" sul terminale dovrebbero incominciare a scorrere numerose scritte. Provate a fare una domanda, ad esempio "Alexa che ora fa?". Se tutto funziona correttamente Alexa dovrebbe rispondere alla vostra richiesta. Se desiderate che l'assistente vocale emetta un suono di conferma di attivazione quando pronunciate la parola Alexa, da terminale dovete schiacciare il tasto "i". Si aprirà il menu

Fig. 12
ReSpeaker 4-Mic
Linear Array Kit.



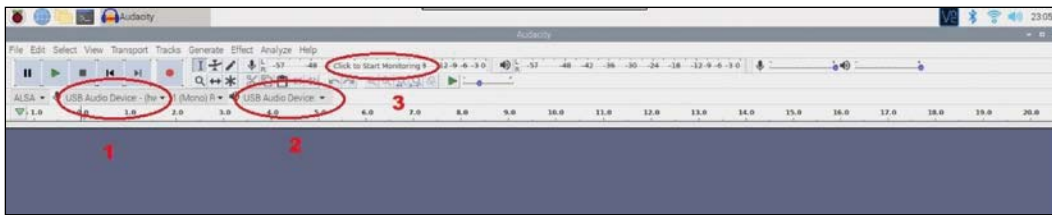


Fig. 13
Impostazioni
dei dispositivi
nel programma
Audacity.

opzioni, selezionate l'opzione "c", successivamente l'opzione "3" e per finire abilitate con il tasto "E". Alexa parla e riconosce molte lingue, per scegliere la lingua di comunicazione in italiano, dal menu opzioni selezionate l'opzione "c" successivamente l'opzione "1" e confermate la lingua italiana "13". Tutte queste impostazioni le potete effettuare anche tramite l'applicazione "Amazon Alexa" che potete installare sul vostro Tablet o Smartphone scaricandola dallo store. Consigliamo di utilizzarla perché da console non avete la possibilità di impostare il fuso orario per gli stati europei.

INSTALLAZIONE DEL PROGRAMMA MAGIC MIRROR 2

Magic Mirror è basato sul framework open-source Electron, funziona grazie a Chromium e Node.js e viene utilizzato per creare applicazioni desktop utilizzando i linguaggi HTML, CSS e JavaScript. Poiché si basa su tecnologie web, il vantaggio principale dell'utilizzo di Electron è quello di essere multiplatforma, consentendo di distribuire le applicazioni con lo stesso codice a tutti i sistemi operativi. Prima di installare il programma *Magic Mirror 2* sulla Raspberry Pi consigliamo di effettuare un controllo degli aggiornamenti del sistema operativo con il comando `"sudo apt-get update"` e successivamente `"sudo apt-get dist-upgrade -y"`. Per chi non è uno smanettone di Linux riportiamo alcuni comandi utili per la manutenzione e l'aggiornamento della vostra Raspberry Pi:

1. `"sudo apt-get update"` scarica la lista delle nuo-

ve versioni aggiornate dei pacchetti disponibili nei *repository*. Questo comando non installa nulla, si limita solo a recuperare le informazioni delle versioni aggiornate.

2. `"apt-get upgrade -y"` scarica e installa le ultime versioni dei pacchetti. I pacchetti attualmente installati con nuove versioni disponibili sono recuperati e aggiornati. In nessun caso vengono rimossi pacchetti attualmente installati.
3. `"sudo apt-get dist-upgrade -y"` scarica e installa le ultime versioni dei pacchetti, le loro dipendenze ed il *kernel* più recente ma non l'avanzamento della versione. Gestisce in modo più intelligente i pacchetti installati.
4. `"sudo do-release-upgrade"` esegue l'avanzamento di versione, passando alla nuova release di Raspbian.
5. `"sudo apt-get clean"` ripulisce la cache locale dei pacchetti.
6. `"sudo apt-get autoremove -y"` rimuove tutti i pacchetti obsoleti e non più necessari.
7. `"lsb_release -d"` visualizzerete la versione corrente del sistema operativo installato sulla Raspberry Pi.
8. `"sudo apt-get purge <package-name> "` rimuove il software specifico conoscendo il nome esatto del pacchetto installato dal sistema.

Terminato l'aggiornamento da console lanciamo il comando `"bash -c '$(curl -sL https://raw.githubusercontent.com/sdeweil/MagicMirror_scripts/master/raspberry.sh)'"` seguendo le istruzioni a monitor. Il programma forse ritornerà dei mes-

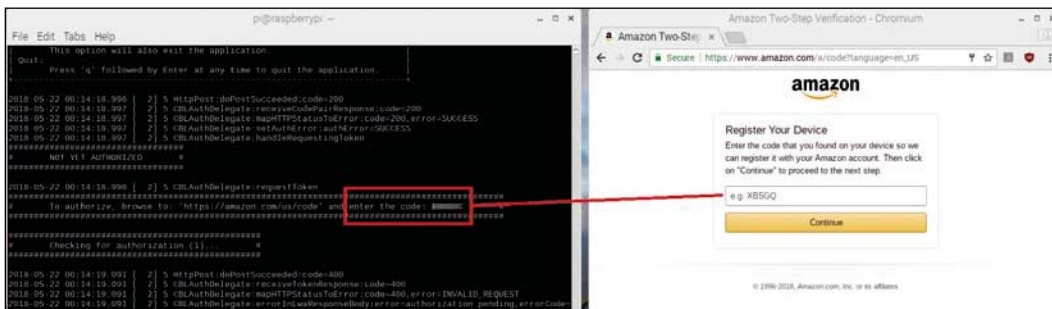


Fig. 14
Ricopiare il
codice di
autenticazione
nella
finestra di
registrazione
di Amazon.

saggi di warning. Niente paura nella cartella `"/install.log"` verranno memorizzati tutti i messaggi di errore del programma che potete andare a leggere successivamente con calma. Se avete già eseguito l'installazione e volete fare solo l'aggiornamento da console potete lanciare il comando `"bash -c \"$(curl -sL https://raw.githubusercontent.com/sdetweil/MagicMirror_scripts/master/upgrade-script.sh)\" apply"`. Il log in questo caso viene salvato nel file `upgrade.log` che si trova nella cartella `"MagicMirror/installers/"`.

Magic Mirror 2 possiede un file di configurazione `config.js` che si trova nella cartella `"/home/pi/MagicMirror/config/"`. Qui potete impostare svariati parametri come la lingua o il formato dell'ora. Per ulteriori informazioni sulle impostazioni vi rimandiamo alla lettura della documentazione che potrete trovare sul sito ufficiale `"https://magicmirror.builders/"`.

Non vi preoccupate se dopo le modifiche del file di configurazione, il programma *Magic Mirror* non si avvia più e non vi ricordate cosa avete modificato prima del blocco. Nella cartella `"/home/pi/MagicMirror/config/"` trovate la copia del file `"config.sample"` con le impostazioni predefinite di configurazione.

La comunità è molto attiva e all'indirizzo `"https://github.com/MichMich/MagicMirror/wiki/3rd-party-modules"` potrete trovare e scaricare parecchi moduli classificati per categorie. Tra i vari moduli abbiamo utilizzato *"MMM-ArduPort"* che abbiamo dovuto modificare per ricevere attraverso la porta USB i valori della temperatura e dell'umidità letti dal sensore HTU21D. Per l'installazione del modulo seguite attentamente le istruzioni riportate nel sito ufficiale `"https://github.com/Dentrax/MMM-ArduPort"`.

Nel **Listato 1** riportiamo la configurazione del modulo *"MMM-ArduPort"* da inserire nel file di configurazione `"config.js"` nella cartella `"/home/pi/MagicMirror/config"`. Sul sito `"http://futura elettronica.it"` potete scaricare tutti i file dei moduli modificati che copierete nelle rispettive cartelle di installazione. Per avviare in automatico il programma *Magic Mirror 2* abbiamo usato il programma `"pm2"`. Per installarlo lanciate da terminale le istruzioni `"sudo npm install -g pm2"`, affinché il programma si avvii dopo il boot lanciate `"sudo pm2 startup"` e per finire create lo script shell di avviamento automatico con il comando `"cd ~"` seguito dal comando `"nano mm.sh"`.

Nel file aperto `mm.sh` scrivete le seguenti righe, nella prima `"cd ~/MagicMirror"` e nella seconda `"DISPLAY=:0 npm start"` salvate e uscite. Assicura-

tevi che il file sia eseguibile lanciando il comando `"chmod +x mm.sh"` e impostate lo script affinché venga eseguito al boot con il comando `"pm2 start mm.sh"`. Verificate il funzionamento riavviando il sistema operativo lanciando il comando `"sudo reboot"`.

Per aggiornare il programma *Magic mirror*, lanciate i seguenti comandi da terminale `"cd /home/pi/MagicMirror"` seguito da `"sudo git reset --hard"`, `"sudo git pull"` e `"sudo npm install"`.

Affinché in futuro non compaia più a monitor la

Listato 1

Configurazione del modulo "MMM-ArduPort".

```

{
  module: 'MMM-ArduPort',
  position: 'top_right',
  header: 'Room temperature',
  config: {
    portname: "/dev/ttyUSB0",
    updateInterval: 1,
    animationSpeed: 1000,
    displayIcons: false,
    showDescription: false,
    hideLoading: false,
    hideWaiting: false,
    useColors: false,
    sensors: [
      {
        name: "Temperature",
        description: "temperature",
        maxValue: 50,
        maxFormat: "({0}°C) VERY HIGH",
        highestValue: 30,
        highestFormat: "({0}°C) HIGH",
        highValue: 15,
        highFormat: "({0}°C) NORMAL",
        lowValue: 10,
        lowFormat: "({0}°C) LOW",
        lowestValue: 5,
        lowestFormat: "({0}°C) VERY LOW",
        minValue: 0,
        minFormat: "({0}°C) OK"
      },
      {
        name: "Humidity",
        description: "humidity",
        maxValue: 50,
        maxFormat: "({0} %) VERY HIGH",
        highestValue: 30,
        highestFormat: "({0} %) HIGH",
        highValue: 15,
        highFormat: "({0} %) NORMAL",
        lowValue: 10,
        lowFormat: "({0} %) LOW",
        lowestValue: 5,
        lowestFormat: "({0} %) VERY LOW",
        minValue: 0,
        minFormat: "({0} %) OK"
      },
    ],
  },
}

```



Fig. 15
Controllate
che alla
locazione
0x68 ci sia
scritto il
valore "UU".

notifica di aggiornamento del Magic Mirror, dovete editare il file `config.js` lanciando da terminale il comando `"sudo nano /home/pi/MagicMirror/config/config.js"` e commentate le seguenti righe:

```
{ module: "updatenotification",
  position: "top_bar"
}
```

Salvate e uscite. Se volete realizzare uno specchio verticale come il nostro, dovete ruotare la visualizzazione dello schermo editando il file di configurazione del sistema operativo Raspbian. Lanciate da terminale il comando `"sudo nano /boot/config.txt"` e aggiungete in fondo al file la seguente riga `"display_rotate=1"`. A titolo informativo diciamo a chi può interessare che in questo file si può anche disabilitare la visualizzazione dello splash screen arcobaleno impostando `"disable_splash = 1"`.

IMPOSTAZIONI RTC

Per aggiungere l'orologio in tempo reale esterno RTC, dovete editare il file `"config.txt"` presente nella cartella `"boot"` lanciando da terminale il comando `"sudo nano /boot/config.txt"` e aggiungete alla fine del file la riga `"dtoverlay = i2c-rtc, ds3231"`. Salvate il file e fate il reboot lanciando il comando `"sudo reboot"`. Quando il sistema operativo si è riavviato loggatevi e lanciate il comando `"sudo i2cdetect -y 1"` e controllate che alla locazione 0x68 ci sia scritto il valore "UU" come visibile in **Fig. 15**.

Ora dovete disabilitare il "fake hwclock" che va ad interferire con il vero hwclock. Lanciate da terminale il seguente comando `"sudo apt-get -y remove fake-hwclock"` confermando affermativamente quando ve lo chiede, successivamente lanciate `"sudo update-rc.d -f fake-hwclock remove"` seguito da `"sudo systemctl disable fake-hwclock"`. Una volta disabilitato il fake hwclock dovete abilitare il vero hardware clock lanciando da terminale `"sudo nano /lib/udev/hwclock-set"` e

togliere il commento alle seguenti linee:

```
#if [ -e /run/systemd/system ] ; then
# exit 0
#fi
...
# /sbin/hwclock --rtc=$dev --systz --badyear
...
# /sbin/hwclock --rtc=$dev systz
```

Quando collegherete la prima volta il modulo RTC alla Raspberry Pi, lanciando il comando `"sudo hwclock -D -r"` noterete che l'ora visualizzata non è esatta. Collegate la RPI alla LAN affinché sincronizzi l'ora da Internet e successivamente con il comando `"date"` controllate se l'ora è esatta. Per scrivere l'ora nel modulo RTC lanciate il comando `"sudo hwclock -w"` e per verificare l'avvenuta scrittura richiamatela lanciando il comando `"sudo hwclock -r"`. Assicuratevi che nel porta batteria del modulo RTC DS3231 sia sempre presente una batteria CR2032 carica, altrimenti spegnendo la RPI l'ora e la data si resetteranno.

Si conclude qui questa prima parte del progetto: completeremo la descrizione dello specchio Smart con l'hardware e la realizzazione pratica nella seconda ed ultima puntata, nel prossimo fascicolo. ■



Cosa occorre?

I componenti utilizzati in questo progetto sono disponibili presso Futura Elettronica. La board Raspberry Pi 3+ (cod. RPI3BPLUS) è in vendita a Euro 49,00, il microfono USB (cod. USBMICMINI) è disponibile a Euro 3,50. 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.



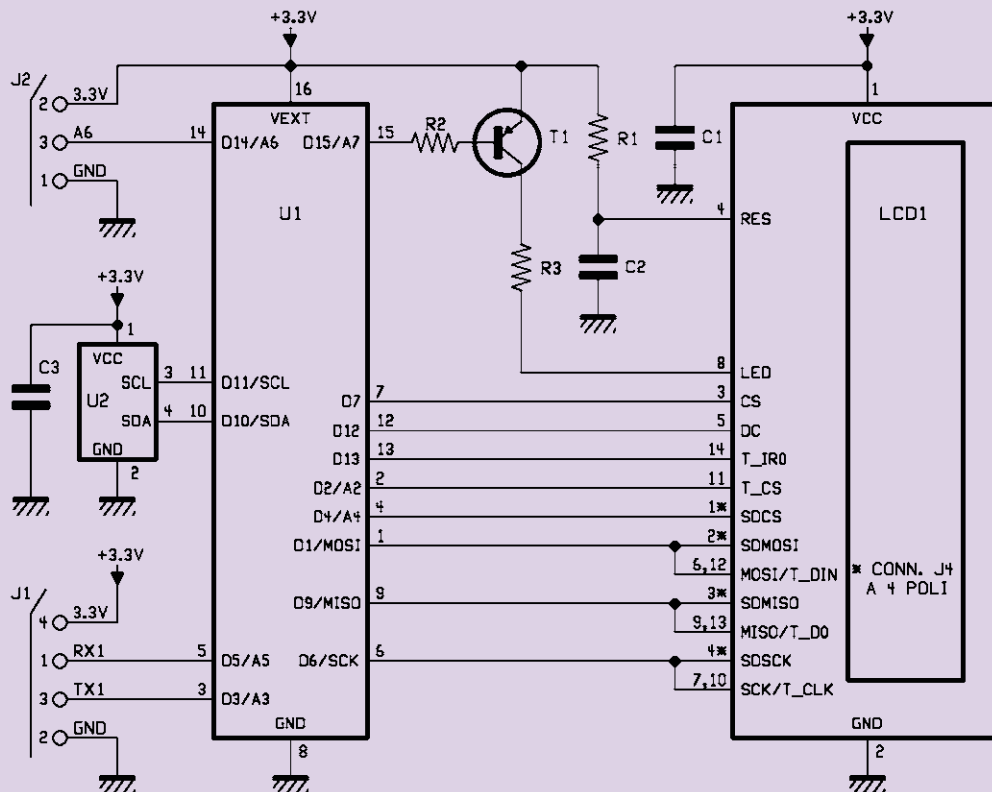
LOGGER GPS CON UCHIP

dell'ing. **GIOVANNI CARRERA**



Crea log delle posizioni rilevate da un ricevitore GPS e ne elabora i dati per calcolare la distanza e la rotta dalla posizione da cui si è partiti, in modo da poter ritornare facilmente al luogo di partenza.

Il modulo uChip è un sistema embedded poco più grande di un circuito integrato DIP a 16 pin, dotato di un microcontrollore Cortex M0 SAMD21-E18A a 32 bit (con RAM di 256 kB) e capace di gestire un display TFT, un touch controller e una scheda SD, che si presta per sviluppare numerose applicazioni che richiedono un'elaborazione di buone prestazioni, un'interfaccia utente e l'interconnessione con vari bus. L'hardware del sistema, con modeste modifiche del software, si presta anche alla gestione di segnali analogici. Nel fascicolo n° 252 avete visto uChip impiegato per realizzare un sistema di monitoraggio ambientale; qui, a riprova della sua versatilità, proponiamo un'applicazione in cui, abbinato a un modulo GPS e una batteria, permetterà di realizzare la localizzazione basata sulla rete di posizionamento satellitare GPS.



Per interfacciarsi con il GPS si utilizzerà l'interfaccia seriale, mentre il *touch display* servirà per interagire col programma e per leggere le numerose informazioni che il ricevitore GPS fornisce e anche la distanza dal punto di partenza del percorso e la rotta che si deve seguire per ritornare. La scheda SD serve per memorizzare le posizioni, dette anche fix, che ogni secondo sono rilevate dal ricevitore. Le coordinate salvate sono convertite in gradi decimali, compatibili con Google My Maps.

Le misure acquisite sulla SD possono essere importate da Excel, usando la virgola come separatore, oppure dal programma Matlab descritto in questo stesso articolo.

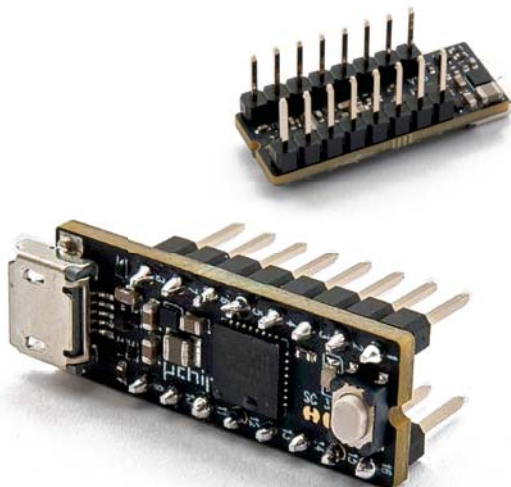
I SISTEMI SATELLITARI DI POSIZIONAMENTO GLOBALE (GNSS)

Il primo prototipo di satellite GPS fu messo in orbita nel febbraio del 1978 e dal 1995 il sistema GPS ha raggiunto la piena capacità operativa per impieghi civili e militari.

Al GPS si sono aggiunti altri sistemi come il russo GLONASS, il cinese BeiDou e l'europeo GALILEO. Inoltre sono stati messi a punto dei sistemi satellitari SBAS (*Satellite Based Augmentation Systems*) per aumentare l'accuratezza di un ricevitore *stand-alone*; questi ultimi utilizzano satelliti geostazionari e una rete di stazioni a terra per eseguire correzioni differenziali a lunga base.

Per spiegare a fondo il funzionamento del GPS occorrono molte pagine, quindi in questo articolo vi forniremo solo i concetti base. Il GPS permette di ricavare la posizione dell'antenna del ricevitore, espressa come latitudine, longitudine e altezza, conoscendo la distanza tra questa ed almeno quattro satelliti o tre se si è a livello del mare.

→ La scheda uChip.



La loro posizione in un certo istante si ricava dai parametri orbitali, detti effemeridi, e la distanza utente-satellite si ricava dal tempo di propagazione dei segnali trasmessi dai satelliti, che viaggiano alla velocità della luce nel vuoto. Come è evidente, la conoscenza del tempo è di notevole importanza, pertanto, oltre alle tre incognite costituite dalle coordinate dell'antenna, c'è anche lo scarto temporale tra gli orologi atomici dei satelliti e quello del ricevitore, molto meno preciso. Quattro equazioni per quattro incognite e, se l'utente è a livello del mare ($h=0$), bastano tre satelliti con una buona geometria in quanto si utilizza il modello ellissoide, che è una figura geometrica regolare che meglio approssima la Terra, in questo caso viene usato l'ellissoide WGS80.

Ogni satellite GPS genera un proprio codice pseudo-casuale PRN (*Pseudo-Random Noise*) di 1.023 bit a una frequenza di 1,023 MHz, cui corrisponde una lunghezza d'onda $\lambda = 293,1$ m, che è la lunghezza spaziale di ogni bit.

I normali ricevitori GPS sono in grado di operare solo sulla frequenza portante $L1 = 1575,42$ MHz mentre quelli più sofisticati operano anche sulla seconda portante $L2 = 1227,60$ MHz e hanno una maggiore precisione.

Il tempo di propagazione si ricava dal processo di aggancio del codice, necessario per la rivelazione del segnale stesso, consistente nel misurare lo scarto tra il segnale PRN, generato localmente dal ricevitore, con quello che viene ricevuto dal satellite. Questo processo si chiama correlazione ed in pratica consiste nel fare scorrere la stringa dei bit del PRN, generato dal ricevitore, fino a farla coincidere con quella del satellite.

L'errore orizzontale di posizione è intorno ai 2-5 m, e dipende da vari fattori, come la geometria



Fig. 1
Il modulo GPS.

dei satelliti, la rifrazione troposferica e ionosferica il rumore del ricevitore e dalla misura di allineamento dei codici. Quello verticale è maggiore e può valere circa il doppio.

Durante la fase della correlazione del codice, il ricevitore varia anche la frequenza MF locale in modo da renderla perfettamente coincidente con quella del satellite; tale processo (*carrier tracking*) porta alla misura della frequenza Doppler che permette di ricavare la **velocità del ricevitore**.

Accuratezze anche inferiori al centimetro si possono raggiungere, sempre in differenziale, con particolari ricevitori che utilizzano anche la fase della portante, oltre ai codici, per cui la lunghezza d'onda passa da 293 metri (dei codici PRN) a 19 cm per la portante $L1$ del GPS.

Di notevole importanza nella determinazione dell'errore è la geometria dei satelliti visti dall'antenna; le loro posizioni ideali sarebbero: uno sulla verticale e gli altri tre equi-spaziati di 120° e bassi sull'orizzonte da potere essere ricevuti dall'antenna e senza troppi ritardi dovuti alla rifrazione iono-troposferica (intorno ai 10° a 15°).

Un ricevitore in grado di elaborare più costellazioni si chiama GNSS, vedendo più satelliti in un certo istante essi possono utilizzare quelli con la geometria migliore il che porta a una maggiore accuratezza del fix.

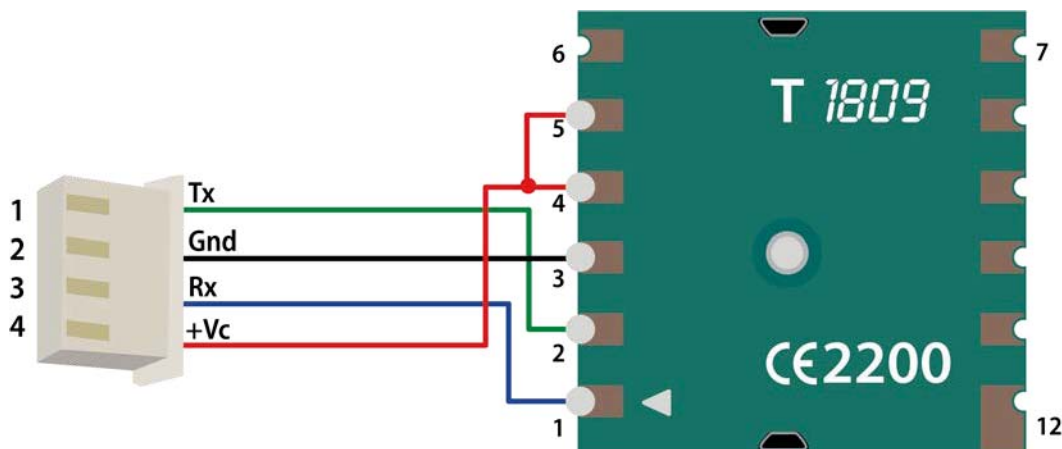


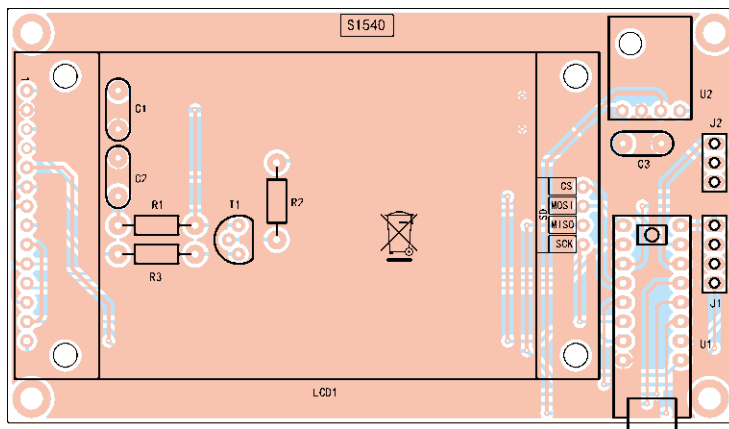
Fig. 2
Collegamenti del GPS.

Elenco Componenti:

R1: 47 kohm
 R2: 1 kohm
 R3: 6,8 ohm
 C1,C3: 100 nF
 C2: 100 nF 63V poliestere
 T1: BC557
 U1: uCHIP
 U2: -
 LCD1: display touchscreen 240x320 2,4"
 J1: strip maschio 4 poli
 J2: strip maschio 3 poli

Varie:

- strip femmina 4 poli
- strip femmina 8 poli
- strip femmina 14 poli
- Circuito stampato S1540 (96x54)



Il ricevitore, oltre alla posizione fornisce anche la velocità dell'utente e il tempo UTC (*Universal Time Coordinated*) che ha la precisione degli orologi atomici usati sui satelliti e del segmento di controllo delle stazioni terrestri.

IL RICEVITORE GPS UTILIZZATO

Per questa applicazione va bene un modulo GNSS con alimentazione e segnali a 3,3V. Quello utilizzato è il ricevitore Quectel modello L80-R, molto compatto ed economico, visibile in **Fig. 1**. Questo ricevitore GPS è in grado di elaborare fino a 66 canali di cui 22 tracciabili simultaneamente e ha montata anche un'antenna *patch* passiva. L'interfaccia seriale può essere programmata con baud rate da 4800 a 115.200 bps (il default è 9600). Le sentenze NMEA 0183 trasmesse sono: GGA, RMC, GSA, GSV, VTG, GLL e TXT. L'alimentazione va da 3 a 4,3 Vcc (quella consigliata è 3,3 V) con un consumo di circa 25 mA. Esso è un modulo adatto per montaggio superficiale ma le piazzole hanno 2,54 mm di passo e si può facilmente saldare, con sottili fili flessibili, al

connettore femmina JST da 4 pin, compatibile con quello del sistema uChip24, come si vede in **Fig. 2**, dove si vede anche la corretta numerazione dei pin. Date le esigue dimensioni dell'antenna passiva (15x15 mm) il modulo si può anche montare su una piastra metallica: una piastrina di circa 70x60 mm per circuiti stampati è la soluzione ideale. Essa funge da *ground plane*, aumentando sensibilmente il guadagno e le prestazioni dell'antenna. Il modulino può essere fissato alla piastra con colla epossidica usando una piastrina di qualche millimetro di spessore per evitare che le saldature, sul lato inferiore, tocchino la piastra metallica. È bene anche saldare un filo tra lo schermo del modulo e la piastra per metterla a massa. Combinando l'AGPS (*Assisted GPS*) avanzato, chiamato **EASY** (*Embedded Assist System*), e la collaudata tecnologia AlwaysLocate, L80-R raggiunge le massime prestazioni e soddisfa pienamente le aspettative. Le numerose prove a punto fisso, ottenute collocando il sistema, in registrazione su SD-Card per qualche decina di minuti, su una posizione fissa, favorevole e sgombra da ostacoli

hanno dato risultati notevoli, anche considerando che esso utilizza soltanto i satelliti GPS e quelli SBAS (EGNOS per l'Europa). In queste condizioni statiche, si è misurato un CEP R95 di soli 50 cm nelle condizioni migliori. Questo test di accuratezza orizzontale rileva il raggio del cerchio contenente il 95% delle misure. Con la dinamica del movimento le cose peggiorano, ma basta fermarsi un poco per avere misure di distanza e angolo più accurate.

I MESSAGGI GPS NMEA

I ricevitori satellitari trasmettono su porta seriale messaggi in formato NMEA (*National Marine Electronics Association*), dette anche sentenze. Ecco un esempio delle sentenze trasmesse dal ricevitore Quectel:

```
$GPGGA,105309.000,4429.2666,N,00907.4267,E,1,6,1.35,
666.1,M,48.0,M,,*5E
$GPGLL,4429.2666,N,00907.4267,E,105309.000,A,A*51
$GPGSA,A,3,30,05,07,28,14,13,,,,,1.66,1.35,0.97*06
$GPGSV,2,1,07,30,65,054,47,05,64,245,46,13,50,299,46
,14,47,135,47*7C
$GPGSV,2,2,07,28,42,143,46,07,32,057,39,08,,15*74
$GPRMC,105309.000,A,4429.2666,N,00907.4267,E,0.03,18
0.62,080121,,,A*62
$GPVTG,180.62,T,,M,0.03,N,0.05,K,A*36
```

Il ricevitore L80-R trasmette le sentenze \$GPGGA, \$GPGLL, \$GPGSA, \$GPGSV, \$GPRMC e \$GPVTG le cui caratteristiche principali sono:

- **\$GPGGA**: orario, posizione 3D e qualità dei dati ricevuti.
- **\$GPGLL**: posizione, orario e stato del fix.
- **\$GPGSA**: modalità operativa del ricevitore GPS, satelliti utilizzati nella soluzione di navigazione riportati dalla frase GGA o GNS e valori DOP.
- **\$GPGSV**: Numero di satelliti (SV) in vista, ID satellite, elevazione, azimuth e valore SNR. Massimo quattro satelliti per trasmissione.
- **\$GPRMC**: ora, data, posizione, rotta e velocità.
- **\$GPVTG**: rotta e velocità relative alla terra.

Le sentenze utili al programma sono la \$GPGGA e la \$GPRMC. La seconda serve per ricavare la data che è impiegata, insieme all'ora, per generare il nome del file che sarà di questo tipo: ddmm yyhh.csv. Se la registrazione dovesse durare più di quell'ora, è creato un nuovo file con l'ora seguente. Il campionamento è eseguito ogni secondo.

GGA - GLOBAL POSITIONING SYSTEM FIX DATA

In questa applicazione sono state utilizzate le sentenze \$GPGGA, che si appoggiano ai satelliti del sistema di localizzazione americano Navstar/GPS.



Tabella 1
Informazioni contenute nella sentenza GGA.

i	CAMPO	NOME	DESCRIZIONE
1	hhmmss.sss	ora UTC	in formato hhmmss.ss
2	llll.llll	latitudine	in formato dddmm.mmmm, inseriti zeri iniziali.
3	a	indicatore N/S	'N' = Nord; 'S' = Sud
4	yyyyy.yyyy	longitudine	in formato dddmm.mmmm, inseriti zeri iniziali.
5	a	indicatore E/W	'E' = Est; 'W' = Ovest
6	x	indicatore qualità fix	0: posizione fix non disponibile 1: posizione fix valida in modo SPS 2: posizione fix valida in modo differenziale GPS
7	uu	N. satelliti usati	(00 ~ 24)
8	v.v	HDOP	Horizontal Dilution Of Precision, (00.0 ~ 99.9)
9	w.www	altitudine	altitudine media sul liv. del mare in metri
10	M		unità di misura dell'altitudine in metri
11	x.x	separazione geoidale	distanza tra la superficie dell'ellissoide e la superficie del geoido, misurata lungo una linea perpendicolare all'ellissoide.
12	M		unità di misura della separazione geoidale in metri
13	zzzz	DGPS Station ID	Identificativo stazione DGPS, NULL quando non DGPS
14	hh	checksum	Controllo errore, preceduto da asterisco



Tabella 2
Informazioni contenute nella sentenza RMC.

i	CAMPO	NOME	DESCRIZIONE
1	hhmmss.sss	tempo UTC	in hhmmss.sss format (000000.000 ~ 235959.999)
2	x	stato	'V' = Navigation receiver warning 'A' = Data Valid
3	llll.lliilll	latitudine	In formato dddmm.mmmmmmm, inseriti zeri iniziali.
4	A	indicatore N/S	'N' = Nord; 'S' = Sud
5	yyyyy.yyyyyyy	longitudine	In formato dddmm.mmmmmmm, inseriti zeri iniziali.
6	A	indicatore E/W	'E' = Est; 'W' = Ovest
7	x.x	velocità sulla terra	Espressa in nodi (000.0 ~ 999.9)
8	u.u	rotta sulla terra	Espressa in gradi (000.0 ~ 359.9)
9	xxxxxx	data UTC	In formato ddmmyy
10	v	Indicatore di modo	'N' = Data not valid 'A' = Autonomous mode 'D' = Differential mode
11	hh	checksum	Controllo errore, preceduto da '*'

Di seguito il formato e in **Tabella 1** tutte le informazioni relative a questa sentenza.

\$GPGGA, hhmmss.ss, llll. lll, a, yyyyy.yyy, a, x, uu, v.v, w.w, M, x.x, M, , zzzz*hh<CR><LF>

Un parametro molto significativo sull'accuratezza della posizione è **HDOP**, la diluizione geometrica orizzontale, esso dipende dalla geometria spaziale dei satelliti usati per il calcolo della posizione (fix). Se $HDOP < 1$ si ha un'ottima misura di posizione, aumentando questo valore si ha un degrado dell'accuratezza della posizione. Ecco perché occorrono molti satelliti in vista per poter scegliere quelli con la migliore geometria.

RMC – RECOMMENDED MINIMUM SPECIFIC GNSS DATA

Contiene l'ora, la data, la posizione, la rotta e la velocità, con il formato:

\$GPRMC, hhmmss.sss, x, llll. lll, a, yyyyy.yyy, a, x.x, u.u, x xxxxx, , , v*hh<CR><LF>

I parametri corrispondenti sono dettagliati nella **Tabella 2**.

CALCOLO DISTANZA TRA DUE PUNTI

Se le misure di posizione riguardano una zona abbastanza limitata del nostro pianeta, si può passare dal modello ellissoide della Terra a quello

Indicando con:

$$\begin{aligned} lat1, lon1 & \text{ espresse in [radianti]} \\ lat2, lon2 & \text{ espresse in [radianti]} \\ \Delta lat &= lat2 - lat1 \\ \Delta lon &= lon2 - lon1 \\ R_T &= 6378137 \text{ [m]} \end{aligned}$$

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \times \cos(lat2) \times \sin^2\left(\frac{\Delta lon}{2}\right)$$

$$d = 2 \times R_T \times \arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \text{ distanza tra i due punti}$$

Fig. 3
Formula ell'emisenverso (haversine).

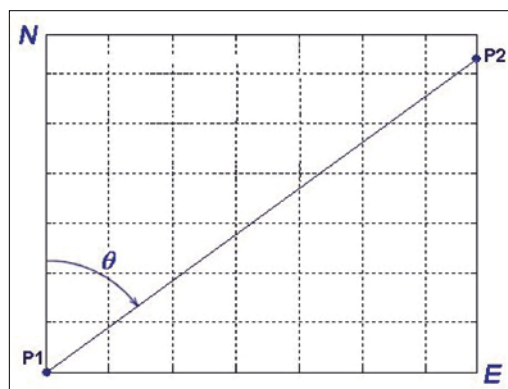


Fig. 4
L'angolo di rotta.

sferico. La distanza più breve tra due punti su una sfera, conosciuta con il termine di ortodromia, è il segmento di un circolo massimo. Sono circoli massimi quei cerchi giacenti in piani che passano per il centro della Terra.

Si sono confrontati due algoritmi per calcolare l'ortodromia, uno è la formula dell'**emiseno verso** o **haversine**, e l'altro è la **legge dei coseni sferici**, più semplice ma meno accurata per distanze sub-metriche. In **Fig. 3** si vede la formula *haversine* per calcolare la distanza tra due punti. La libreria TinyGPSPlus usa proprio questa formula, con la seguente funzione:

```
Dist = gps.distanceBetween(Latm, Longm, Latd, Longd);
```

Tutti i calcoli che elaborano le coordinate devono essere fatti in doppia precisione (8 byte in virgola mobile) e non possono essere fatti da MCU a 8 bit come quelle usate da Arduino Uno, Mega o simili. Per cui occorre una MCU a 32 bit come Arduino Due, Arduino Zero, ESP8266 (meglio ESP32) o uChip che usa un SAMD21 come Arduino Zero. Da simulazioni si è verificato che in singola precisione gli errori si sono verificati soprattutto per il calcolo dell'angolo. La libreria TinyGPSPlus utilizza la doppia precisione.

CALCOLO DELL'ANGOLO DI ROTTA TRA DUE PUNTI

La rotta, è l'angolo che la retta che congiunge i due punti forma con il meridiano. Essa è espressa in gradi crescenti in senso orario da 0° a 360°, come si vede nell'esempio in **Fig. 4**. Le coordinate dei due punti sono state convertite in coordinate UTM (*Universal Transverse Mercator*) per realizzare il grafico, con assi in metri relativi alla posizione P1, ma l'angolo è stato calcolato con la formula proposta. Se il punto P2 giacesse sul meridiano, sopra P1, la rotta sarebbe di 0° o di 360° gradi, se fosse sotto P1, la rotta sarebbe di 180°.

La formula per calcolare la rotta iniziale che, se seguita in linea retta lungo un circolo massimo, porterà dal punto $P_1(\varphi_1, \lambda_1)$ al punto $P_2(\varphi_2, \lambda_2)$ è:

$$\theta = \text{atan2}(\sin \Delta\lambda \cdot \cos \varphi_2, \cos \varphi_1 \cdot \sin \varphi_2 - \sin \varphi_1 \cdot \cos \varphi_2 \cdot \cos \Delta\lambda) * 180/\pi$$

Dove $\Delta\lambda = \lambda_2 - \lambda_1$. Queste formulazioni non tengono in considerazione gli ostacoli, quindi la rotta iniziale si modificherà man mano che si aggirano gli ostacoli.

La formula fornisce angoli compresi tra 0° e *180°,

per avere l'angolo di rotta compreso tra 0° e 360° basta sommare 360° ai soli valori negativi.

La libreria TinyGPSPlus ha questa funzione per calcolare la rotta:

```
Course = gps.courseTo(Latd, Longd, Latm, Longm);
```

USARE IL SISTEMA PER ORIENTARSI

Appena acceso lo strumento, occorre posare il sistema o tenerlo fisso per almeno un paio di minuti, in modo che possa iniziare ad agganciare i satelliti e calcolare le coordinate medie del punto di partenza con un certo numero di misure a punto fisso (60).

Quindi inizia a calcolare la distanza e la rotta. Tutte queste fasi sono ben identificate sul display.

Oltre a questo sistema è necessaria una bussola con ghiera graduata tipo quella visibile nella **Fig. 5**. Occorre ruotare la ghiera fino a fare coincidere il suo Nord (0° = 360°) con la lancetta rossa che indica il Nord magnetico.

Quindi basta leggere la rotta indicata dallo strumento (bearing) e seguire la direzione indicata



Fig. 5
La bussola.

dalla ghiera. Le misure di posizione hanno un errore orizzontale che, nelle condizioni migliori, è dell'ordine di 2,5 m, anche se nelle prove a punto fisso questo GPS ha dato ottimi risultati. Questo vuol dire che la misura della distanza e dell'angolo di rotta possono avere errori notevoli se si è nelle immediate vicinanze. Volendo, con qualche modifica al programma si possono impostare le coordinate del punto di arrivo per poterne calcolare la distanza e la direzione da prendere, esattamente come si fa a bordo per determinare la rotta da prendere per arrivare a destinazione. A differenza del mare aperto, in terraferma le cose sono più problematiche per via dei numerosi ostacoli, quale l'orografia del territorio, gli edifici, gli alberi e le persone stesse. Questi sono ostacoli sia per la ricezione diretta dei satelliti sia per l'utente che non può prendere una direzione qualsiasi. Se ci troviamo in un fitto bosco occorre aspettare e fare la misura in una radura o in una posizione più elevata che permetta di vedere un'ampia porzione di cielo. La distanza non tiene conto della quota e non è la lunghezza del cammino che si deve compiere ma è quella in linea d'aria.

Naturalmente questi concetti servono per tutti i tipi di ricevitori GNSS e quelli più precisi, come quelli che operano anche sulla fase della portante, richiedono ancora maggiori precauzioni.

REALIZZAZIONE PRATICA

Per la realizzazione del prototipo, visibile nelle **Fig. 6** e **Fig. 7** e in quella introduttiva, abbiamo realizzato un apposito guscio con stampante 3D, dove alloggiare il PCB, il ricevitore GPS, la batteria tipo 18650 completa di porta-batteria e l'interruttore. L'elettronica è quella del sistema proposto nell'articolo "MONITORAGGIO AMBIENTALE con uChip" del numero 257 più il modulo ricevitore Quectel L80R (Cod: 2846-GPSL80R). Per comodità riportiamo in queste pagine il piano di montaggio della scheda madre cui saranno applicati la uChip e il display, della quale è possibile scaricare dal nostro sito www.elettronica.in.it i file per la realizzazione del circuito stampato.

Inciso e forato il PCB, saldati i suoi componenti, si possono innestare il modulo uChip e il display grafico. Si collega quindi il modulo ricevitore GPS. La batteria ai polimeri di litio da 3,7 V potrebbe essere piatta da (1100 mAh, cod: 7893-

Fig. 6
Il prototipo, vista frontale.



Fig. 7
Il sistema nel suo contenitore.



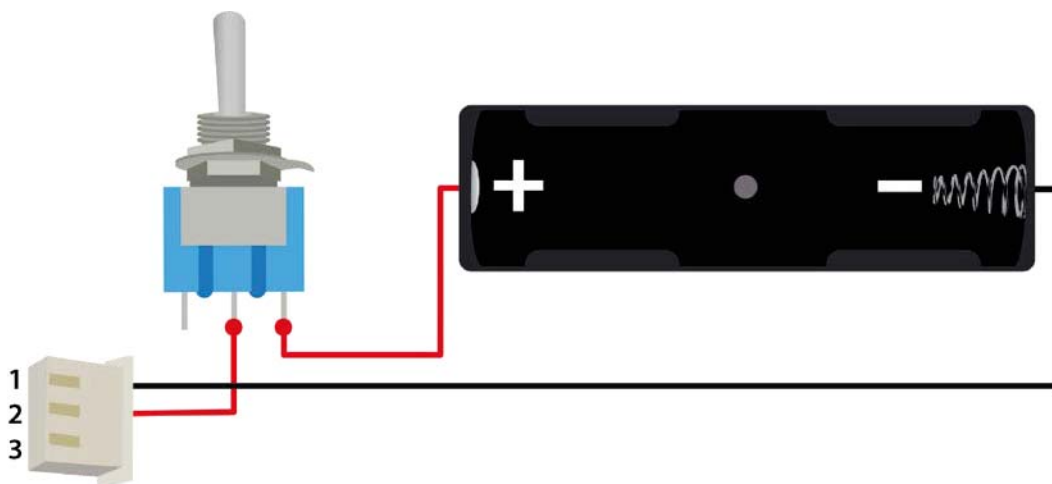


Fig. 8
Alimentazione
del sistema.

PLIB1100) oppure tonda tipo 18650 da 2200 mAh (Cod: 8240-18650) con porta-batteria (Cod: 1606-1X18650) in entrambi casi si deve saldare un connettore femmina JST da 3 pin e un interruttore.

In **Fig. 8** è visibile la versione con porta-batteria per batteria in formato 18650, si noti la corretta numerazione del connettore JST.

Per questa applicazione, l'alimentazione è sul pin 16 (Vext) di uChip, che è anche collegato al pin 2 del connettore. Al fine di evitare danni irreparabili, occorre fare molta attenzione a come si montano i connettori sul circuito stampato. La **Fig. 9** ne mostra i corretti orientamenti.

Si rimanda all'articolo citato per la realizzazione del circuito stampato, per il montaggio dei componenti sulla scheda e per l'upload dell'attuale programma su uChip.

La cosa migliore è quella di realizzare un involucro ad hoc con l'uso di una stampante 3D, ma in assenza di essa, si può facilmente realizzare con pannelli di plastica Forex da 5 mm di spessore, incollati con colla per PVC, come si è fatto per il prototipo.

LA LIBRERIA TINYGPSPLUS

Esistono in rete diverse librerie per estrarre le informazioni dalle sentenze NMEA, una delle più note è TinyGPS di Mikal Hart. Questa libreria era nata per funzionare su una Arduino UNO, che è basata su una MCU a 8 bit, inadeguata per fare calcoli in doppia precisione, e che fornisce le coordinate con la risoluzione di un milionesimo di grado, in formato *long*. Si è scelta la versione TinyGPSPlus la cui interfaccia di programmazione è più intuitiva e le coordinate sono fornite anche in doppia precisione. Nel nostro caso, la MCU di uChip è a 32 bit ed elabora in "double" senza problemi di velocità

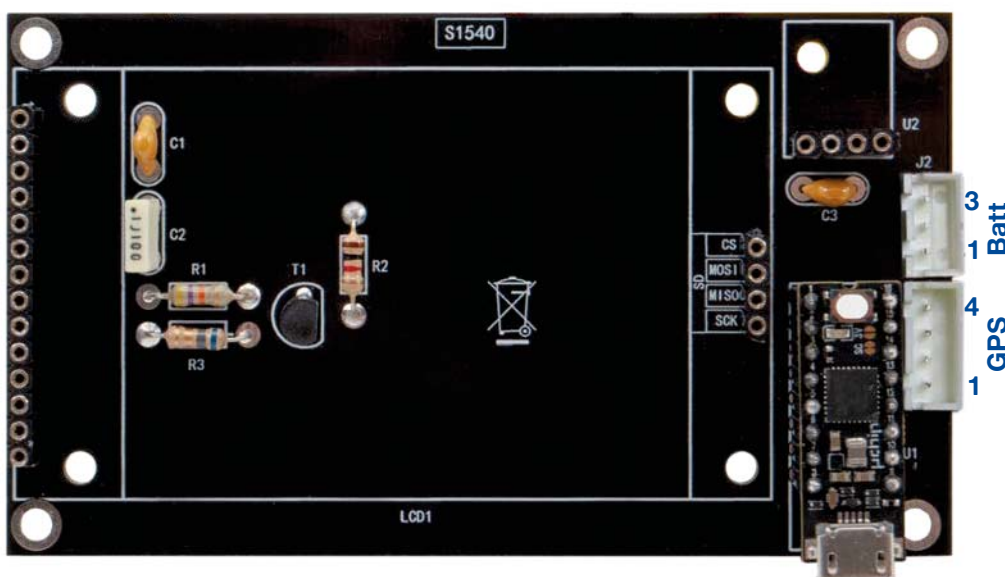


Fig. 9
Disposizione
dei connettori
per il GPS e
per la batteria.

di accuratezza. Entrambe le librerie estraggono la posizione di base, l'altitudine, la rotta, l'ora e la data, ecc. dalle due sentenze NMEA GGA e RMC. Il cuore della libreria consiste nel leggere le due sentenze NMEA ed estrarre e decodificare le informazioni principali come la posizione, la data e l'ora, ecc. La latitudine e la longitudine, originalmente espresse in gradi e primi, sono convertite in gradi decimali e, per il calcolo della distanza e dell'angolo di rotta sono ulteriormente convertite in radianti, che sono gli argomenti delle funzioni trigonometriche. Unico neo è che la libreria ignora il *quality index*, presente nelle sentenze GGA, indispensabile per vedere che tipo di fix si ha, esso deve essere pari a 1 o superiore (per altri tipi di ricevitori). Ma esiste un modo per estrarre un qualsiasi parametro contenuto nelle sentenze. In questo caso l'indicatore di qualità è nel campo 6 delle sentenze GGA, per cui:

```
TinyGPSCustom Qind(gps,"GPGGA",6);
```

La data e l'ora che in NMEA sono stringhe, sono trasformate dalla libreria in numeri, cosa alquanto curiosa in quanto sono grandezze non decimali, per cui risultano illeggibili. Ad esempio il tempo (*time*) è espresso in GGA come hhmmss.cc: se l'ora è

000100.00 la libreria restituisce il numero 10000, che è di difficile interpretazione anche perché sono stampati sul display e sul file come stringhe. Per cui si sono aggiunte le due istruzioni:

```
TinyGPSCustom Times(gps,"GPGGA",1);
// extract time (char array)
TinyGPSCustom Dates(gps,"GPRMC",9);
// extract date (char array)
```

Queste restituiscono le stringhe come *char array*. In qualsiasi momento si può esaminare se un oggetto è valido o aggiornato. La funzione **isValid()** dirà se l'oggetto contiene dati validi ed è sicuro da interrogare. Allo stesso modo, **isUpdated()** indica se il valore dell'oggetto è stato aggiornato (non necessariamente cambiato) dall'ultima volta che è stato interrogato. Infine, se si vuole sapere quanto sono obsoleti i dati di un oggetto, si chiama la funzione **age()**, che restituisce il numero di millisecondi dall'ultimo aggiornamento. Se questo restituisce un valore maggiore di 1500 circa, potrebbero esserci problemi di perdita di dati.

IL PROGRAMMA

All'avvio, il programma controlla che sia presente la scheda SD, se essa è presente è abilitata l'eventuale acquisizione (SDok=true) delle misure su file,

↓ Listato 1

```
void setup(void) {
  uChipSetVextValue(VEXT_3V3);
  pinMode(TFTLed, OUTPUT); // TFT backlight (ON = LOW)
  ts.begin(); // initialize XPT2046_Touchscreen
  if (!SD.begin(SD_CS)) {
    SDok = false;
  } else SDok = true;
  tft.begin();
  digitalWrite(TFTLed, LOW); // turn ON display leds
  tft.fillScreen(ILI9341_BLACK);
  tft.setTextColor(ILI9341_CYAN);
  tft.setTextSize(2);
  tft.setCursor(0,50);
  tft.print("uChip GNSS logger");
  tft.setCursor(0,84);
  if (SDok) tft.print("SD initialised!");
  else {
    tft.setTextColor(ILI9341_RED);
    tft.print("SD card failed!");
  }
  delay(2000);
  ok = false;
  Serial1.begin(BaudRate); // serial port for GPS interface
  tft.setCursor(0,124);
  tft.setTextColor(ILI9341_WHITE);
  tft.println("Waiting for valid");
  tft.print("GPS data .....");
} // end of setup
```



Listato 2

```
void GetData() {
    String Qis = Qind.value();
    Qi = Qis.toInt();
    if (Qi > 0) fixok= true;
    else fixok= false; // not valid data
    Latd = gps.location.lat(); // Latitude in degrees (double)
    Longd = gps.location.lng(); // Longitude in degrees (double)
    h = gps.altitude.meters(); // Altitude in meters (double)
    hhmss = Times.value(); // time in HHMMSS.CC String
    hhmss = hhmss.substring(0,6); // only hours, minutes and seconds
    HDOP = gps.hdop.value()/100.0; // Horizontal Dilution of Precision
    nsat = gps.satellites.value(); // Number of satellites in use
    if (SDok && fixok) { // generate the file name based on date and hour
        String hhs = hhmss.substring(0,2); // only hours
        String ddmdd = Dates.value();
        filename = ddmdd+hhs+".CSV"; // the csv file name is the date and hour
    }
    if (nm < NuAv && fixok) { // calculates the average of the initial fix
        Latm += Latd;
        Longm += Longd;
        nm++;
        if (nm == NuAv) {
            fixmean= true;
            Latm /= NuAv;
            Longm /= NuAv;
        }
    }
    if (SDok && fixok && Acqf) {
        File dataFile = SD.open(filename, FILE_WRITE);
        if (dataFile) { // if the file is available, write to it:
            dataFile.print(Latd,7);
            dataFile.print(c);
            dataFile.print(Longd,7);
            String outdata = c+hhmss+c+String(nsat)+c+String(HDOP,1)+c+String(h,1); // other data to record
            dataFile.println(outdata);
            vep++;
            dataFile.close();
        }
    }
    DisplayData();
}
}
```

altrimenti esse sono solo visualizzate sul display. Se la SD non è inserita o se è guasta, stampa sul display "SD card failed!". La MCU di uChip ha una seriale hardware (Serial1) oltre a quella USB, per cui non si è usata quella software. Il baud-rate del nostro GPS, come quello della maggior parte dei moderni moduli in commercio, è di 9600 baud. Se si ha un modulo GPS a 4800 o altro, basta modificare la variabile BaudRate. L'unica cosa da controllare con cura è il livello dei segnali seriali che devono essere di 3,3V e anche l'alimentazione.

Il **setup()** termina con la stampa della scritta "Waiting for valid GPS data" per indicare che è in attesa di dati validi (vedi **Listato 1**). La libreria funziona carattere-per-carattere ricevuto dal GPS e letto dalla funzione encode:

```
while (Serial1.available() > 0){
```

```
    gps.encode(Serial1.read());
```

Per sincronizzare le misure si deve verificare che siano tutte aggiornate le variabili che ci servono:

```
if (gps.location.isUpdated() && Qind.isUpdated()
    && gps.altitude.isUpdated() && gps.satellites.
isUpdated()
    && gps.hdop.isUpdated() && Times.isUpdated() &&
Dates.isUpdated())
{
    GetData();
}
```

La funzione **GetData()** estrae ed elabora tutte le informazioni utili al programma (vedi **Listato 2**). Le coordinate sono contenute in **gps.location**, e sono lette in gradi decimali, in formato double. Il programma attende che l'indicatore di qualità **Qi** sia maggiore di zero, per questo ricevitore esso

vale uno quando ci sono almeno tre satelliti, altrimenti vale zero.

All'accensione il ricevitore impiega alcune decine di secondi per iniziare a identificare i satelliti in vista. Dopo aver ricevuto i primi dati validi, il programma esegue la media di **NuAv** fix, questo è necessario per avere una buona stima delle coordinate del punto di partenza.

```
if (nm < NuAv && fixok){  
  // calculates the average of the initial fix  
  Latm += Latd;  
  Longm += Longd;  
  nm++;  
  if (nm == NuAv){  
    fixmean= true;  
    Latm /= NuAv;  
    Longm /= NuAv;  
  }  
}
```

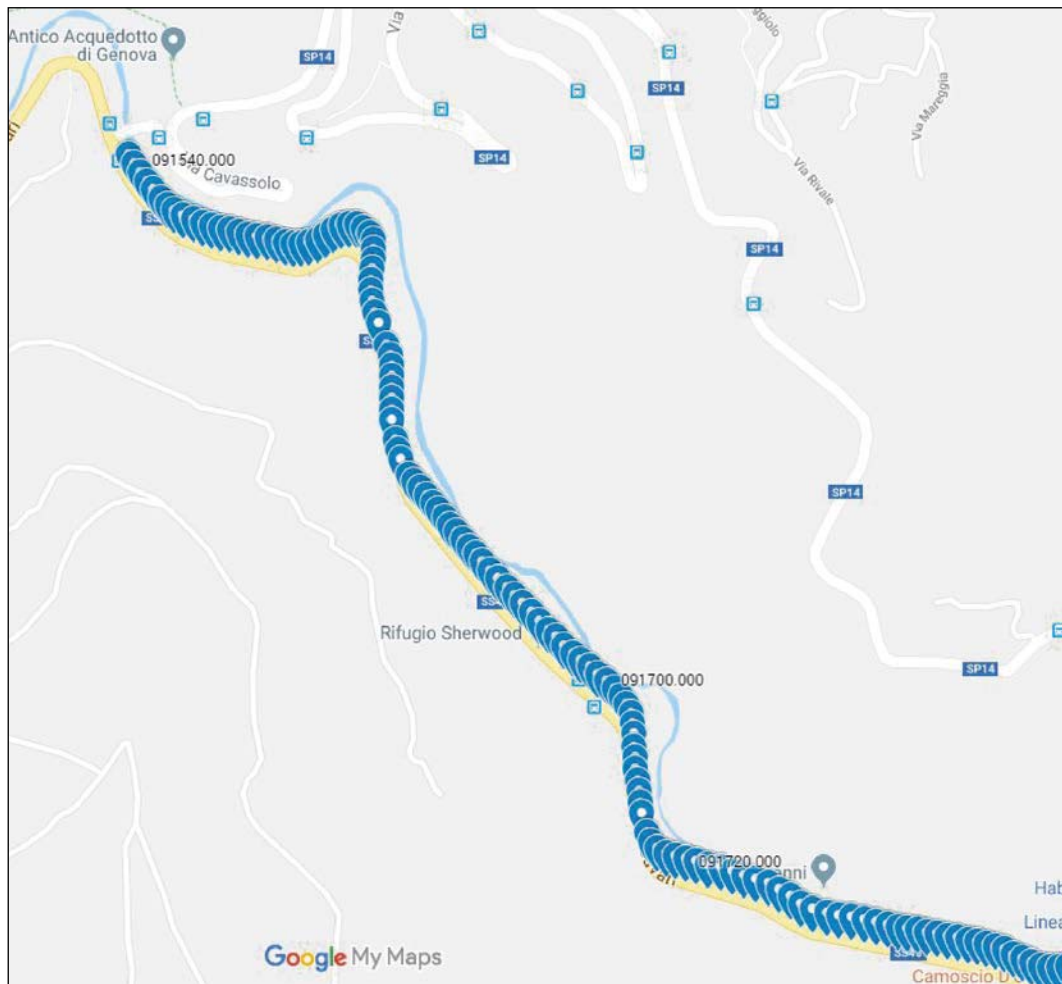
Durante questa fase si deve lasciare ferma l'antenna e assicurarsi che non ci siano ostacoli nelle vicinanze. Sul display comparirà la scritta "N means = xx", dove xx è il numero campioni mediati fino a quel momento. Nel programma si è posto **NuAv=60** pari a un minuto.

Quindi, **fixmean= true** e sul display compariranno altre due righe con le coordinate medie del punto di partenza. Se presente la SD e se i dati sono validi, è generato il nome del file in base alla data e l'ora:

```
if (SDok && fixok) {  
  // generate the file name based on date and hour  
  String hhs = hhmss.substring(0,2); //only hours  
  String ddmdd = Dates.value();  
  filename = ddmdd+hhs+".CSV";  
  // the csv file name is the date and hour  
}
```

Le misure sono stampate sul display ogni secondo

➔ **Fig. 10**
Esempio di
tracciamento
del percorso.



▼ Listato 3

```
% program GPSElab.m legge i dati scritti da uChipGPS:
% latitudine, longitudine, ora UTC, nsat, HDOP, altitudine
% Giovanni Carrera 07 maggio 2021
clear
clc
files= dir('*.csv');
nfile=size(files,1);% numero file trovati
for indf=1:nfile,
    fprintf('%d) %s\n',indf,files(indf).name);
end
while true
    nf = input('numero di file da leggere = ');
    if (nf>=1) && (nf<=nfile), break, end
end
nomefile = files(nf).name;
fid = fopen(nomefile);
if fid==-1
    error([nomefile ' : il file non esiste']);
end
[D] = textscan(fid,'%f %f %s %d8 %f %f','delimiter',' ','');
fclose(fid);
lat = D{1,1};
long = D{1,2};
ftime = D{1,3};
nsat = D{1,4};
HDOP = D{1,5};
h = D{1,6};
ne= size (h,1);
time= zeros(1,ne);
for i=1:ne,
    tim = ftime{i};
    hh = str2num(tim(1:2));
    mm = str2num(tim(3:4));
    ss = str2num(tim(5:6));
    time(i) = hh*3600+mm*60+ss; % time in seconds
end
t = time-time(1);
clc
disp('Elaborazioni misure uChipGPS a punto fisso')
fprintf('Nome file elaborato      = %s\n',nomefile);
fprintf('UTC time iniziale = %s e finale = %s\n',ftime{1},ftime{ne});
fprintf('Medie elaborate su        = %d epoche\n',ne);
fprintf('HDOP media                  = %5.2f\n',mean(HDOP));
fprintf('Numero medio satelliti      = %4.1f\n',mean(nsat));
figure(1)
la = lat-min(lat);
lo = long-min(long);
subplot(3,1,1), plot(t,la,'r','LineWidth',2),grid,xlabel('time [s]'),
ylabel('latitudine'),title(['GNSS elaborazioni file: ' nomefile])
subplot(3,1,2), plot(t,lo,'g','LineWidth',2),grid,ylabel('longitudine')
subplot(3,1,3), plot(t,h,'b','LineWidth',2),grid,ylabel('altitudine')
figure(2)
subplot(2,1,1), plot(t,nsat),grid,ylabel('Numero satelliti')
subplot(2,1,2), plot(t,HDOP),grid,ylabel('HDOP'),xlabel('secondi')
```

dalla funzione **DisplayData()**.

Se la scheda SD è inserita ed è valida, sullo schermo è presente il tasto (blu) per avviare (da blu a rosso) o fermare (da rosso a verde e poi blu) l'acquisizione. Quando acquisisce, sul display viene stampato anche il numero di campioni.

Se è presente la SD e il fix è valido e se è stata avviata l'acquisizione salva i dati su un file CSV. Il file CSV (*Comma-Separated Values*) è un file ASCII con i dati separati da virgole. I dati memorizzati dal

sistema uChipGPS hanno il seguente formato:

latitudine, longitudine, ora UTC, nsat, HDOP, altitudine

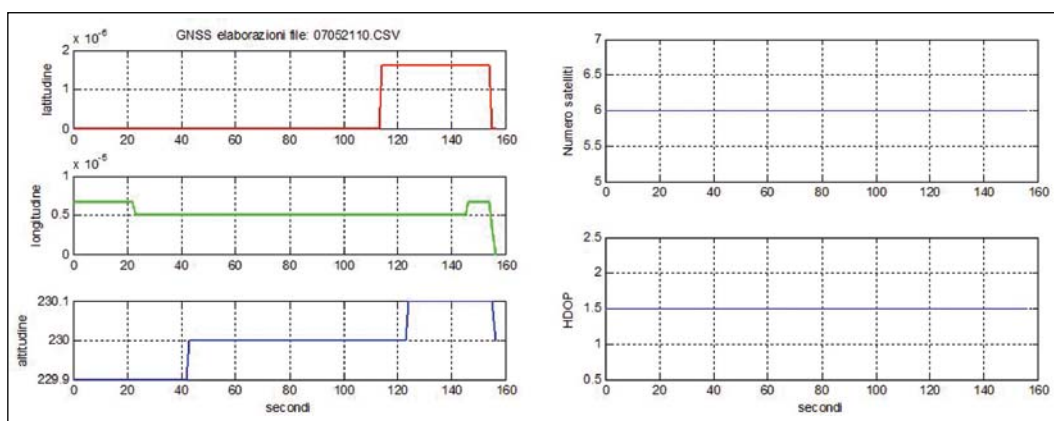
Un esempio è questo:

44.7350017,8.8303500,101535,6,1.5,229.9

con latitudine e longitudine in **gradi decimali**; l'ora UTC ha il formato hhmmss e l'altitudine è espressa in metri. Questo formato è perfettamente compa-



Fig. 11
Grafici di una breve registrazione a punto fisso.



tibile con il programma *My Maps* di Google, in questo modo si possono vedere i punti del percorso sovrapposti alla mappa.

Per importare le informazioni nella mappa **Fig. 10** occorre seguire questi semplici passaggi:

- sul computer, accedere a My Maps (<https://mymaps.google.com/>);
- aprire o creare una mappa;
- nella legenda della mappa, fai clic su "Aggiungi livello";
- date un nome al nuovo livello;
- nel nuovo livello, fai clic su "Importa";
- scegliere o caricare il file `hmmss.csv` e fare clic su "Seleziona";
- selezionare la colonna contenente le latitudini (la prima), le longitudini (la seconda) e le etichette per ogni punto (per esempio l'ora, la terza colonna).

IL PROGRAMMA MATLAB

Questo programma è in grado di elencare tutti i file presenti nella cartella di lavoro, quindi legge il file corrispondente al numero che è immesso. I dati sono messi nella *cell array* `D` dalla funzione:

```
[D] = textscan(fid,'%f %f %s %d8 %f\n', 'delimiter', ',');
```

Quindi sono estratte tutte le informazioni e sono eseguiti i grafici delle coordinate, del numero di satelliti elaborato e della diluizione geometrica orizzontale HDOP. Per motivi grafici la latitudine e longitudine sono riferite al minimo valore misurato. Il programma è riportato nel **Listato 3**. Ecco un esempio di uscite per un'acquisizione a punto fisso

```
Elaborazioni misure uChipGPS a punto fisso
Nome file elaborato      = 07052110.CSV
UTC time iniziale = 101505 e finale = 101741
Medie elaborate su      = 157 epoche
```

```
HDOP media      = 1.50
Numero medio satelliti = 6.0
```

Come si vede in **Fig. 11**, la latitudine è variata di circa 1,7 milionesimi di grado, la longitudine di circa 7 milionesimi e l'altitudine di soli 20 centimetri. Sono misure estremamente precise, quasi anomale, ma che si sono avute anche in diverse altre acquisizioni fatte con questo ricevitore. Infatti, un incremento di un milionesimo di grado di latitudine, per quella zona, corrisponde a circa 11 cm verso Nord e a circa 9 cm verso Est per la longitudine. Questo non significa che il ricevitore ha un'accuratezza sub-metrica, ma che implementa un particolare algoritmo che migliora notevolmente le prestazioni a punto fisso. Bene, con questo abbiamo concluso: termina qui la descrizione di un progetto che mostra la versatilità e la potenza dell'hardware *uChip* abbinato alla nostra scheda madre e a periferiche come un display grafico e un comune ricevitore GPS. □



Cosa occorre?

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

La scheda *uChip* è disponibile con gli strip da saldare (cod. *UCHIP*) a Euro 19,00 e con gli strip già saldati (cod. *UCHIPWSH*) a Euro 19,90. Il display touch da 2,4" (cod. *LCDTOUCH24SER*) è disponibile a Euro 16,90, il GPS (cod. *GPSL80R*) è in vendita a Euro 13,00, il porta batteria per 18650 (cod. *1X18650*) costa Euro 1,00, la batteria da 2200mAh (cod. *18650*) è disponibile a Euro 4,40. 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>

Temperatura sempre sotto controllo!

Termometri a infrarossi senza contatto



€ 38,00
Cod. TP213N

Termometro IR con puntatore LASER (da -20°C a +400°C)

Termometro a infrarossi UNI-T con puntatore LASER per la misura a distanza della temperatura (da -20°C a +400°C). Dispone di display LCD con retroilluminazione, possibilità di visualizzazione in gradi centigradi o in gradi Fahrenheit, spegnimento automatico. La lettura della temperatura è istantanea. La confezione comprende: il termometro e due batterie da 1,5 V tipo AAA.

Termometro IR con puntatore LASER (da -32°C a +420°C)

Termometro a infrarossi UNI-T con puntatore LASER per la misura a distanza della temperatura (da -32°C a +420°C). Dispone di display LCD con retroilluminazione, possibilità di visualizzazione in gradi centigradi o in gradi Fahrenheit, spegnimento automatico. La lettura della temperatura è istantanea (250 ms) con la memorizzazione della soglia minima e massima in caso di scansione continua e prolungata. Memorizzazione fino a 5 set di valori. La confezione comprende: il termometro, la pila da 9 V e la borsa da cintura per il trasporto.

€ 78,00
Cod. TP224



Prezzi IVA inclusa.

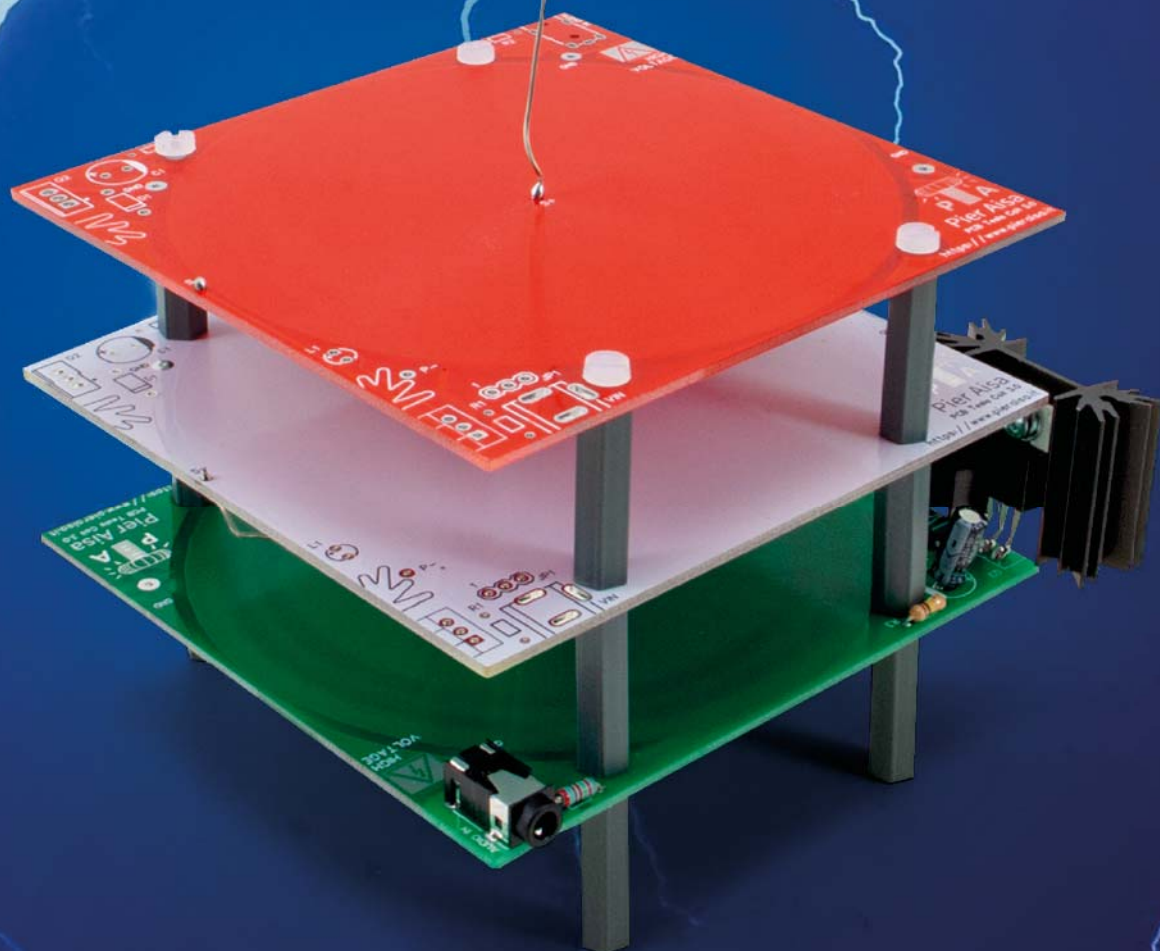


€ 19,00
Cod. FR811

Termometro IR per la lettura della temperatura corporea o di superfici (da +34°C a +42,9°C)

Dispone di ampio display a LED retroilluminato che mostra la lettura in gradi centigradi o Fahrenheit (selezionabile). Gli infrarossi consentono al termometro di leggere la temperatura corporea a una distanza di 5-10 cm in meno di un secondo. L'apparecchio si spegne dopo 8 secondi di standby. Il termometro ha una precisione del display di 0,1°C con un errore massimo di $\pm 0,2^\circ\text{C}$ e un range $34,0^\circ\text{C} \sim 42,9^\circ\text{C}$. Dispone inoltre di segnalazione di allarme con soglia impostabile, cambio di colore del display per indicare le soglie di temperatura misurate. Offre la possibilità di memorizzare fino a 32 misurazioni.





TESLA COIL SU CIRCUITO STAMPATO

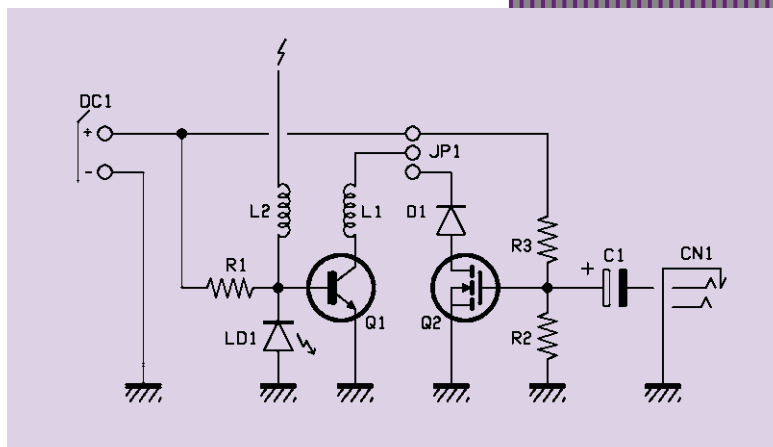
Costruiamo una bobina di tesla realizzandone l'avvolgimento secondario su PCB, così da risparmiarci l'onere dell'avvolgimento manuale.

di PIER-ALESSANDRO AISA

La bobina di Tesla, anche detta tesla-coil, è un dispositivo affascinante ma la cui realizzazione è talvolta ostacolata dal fatto che la realizzazione dell'avvolgimento secondario dell'autotrasformatore è complessa, faticosa ed inoltre richiede una grandissima precisione, considerato l'elevato numero di spire richieste per il funzionamento. Tipicamente questi avvolgimenti sono costituiti da diverse centinaia o migliaia di spire, realizzate con filo di rame smaltato, usato per la costruzione dei trasformatori ed il minimo errore durante la costruzione dell'avvolgimento può compromettere il regolare funzionamento con la nascita di archi elettrici indesiderati fra spira e spira. Questo progetto fa uso di un circuito stampato che realizza l'avvolgimento di secondario in maniera controllata, il che implica il notevole vantaggio di determinare con certezza i parametri elettrici, primari e parassiti, che condizionano in maniera importante il buon funzionamento della bobina, visto che l'operazione è eseguita come lo sbroglio di un circuito stampato tramite un software CAD CAE. Inoltre, siccome l'avvolgimento di secondario è l'elemento più difficile e critico da realizzare, con questo circuito stampato chiunque sarà in grado di mettere in funzione una bobina di Tesla, con l'aggiunta di pochissimi componenti. Infatti è stato selezionato uno schema elettrico molto semplice, che va sotto il nome di "Slayer exciter", che fa uso di un solo transistor come oscillatore. Con questo progetto si possono sperimentare a basso costo, con semplicità ed in sicurezza i fenomeni quali: l'effetto corona, la scarica in aria con arco elettrico e la misura di un campo elettrico, grazie semplicemente all'uso di una lampadina per poterlo visualizzare. Il progetto è accessibile a tutti, anche ai meno esperti in quanto la potenza e la frequenza in gioco non costituiscono un elemento di grande pericolo, come avviene invece per le bobine di Tesla tradizionali, ma bisogna prestare comunque attenzione, perché le tensioni prodotte si aggirano sui 20.000 V, con poca corrente disponibile (inferiore a 1mA).

SCHEMA ELETTRICO

Il principio di funzionamento si basa su l'auto-oscillazione di un transistor, che si instaura tramite l'induzione magnetica che avviene fra l'avvolgimento primario e l'avvolgimento secondario, creando una retroazione sul comando del transistor stesso. Il circuito elettronico si compone di due sezioni: la sezione oscillante e la sezione di modulazione audio.



La sezione oscillante, meglio descritta dallo schema elettrico in queste pagine, è molto semplice: si compone di un transistor BJT di tipo NPN siglato Q1, di una resistenza R1 usata per polarizzare la sua base e di un diodo led LD1, per avere evidenza visiva dell'oscillazione e per bloccare eventuali sovratensioni.

Appena viene fornita alimentazione, il transistor Q1 fa circolare corrente nella bobina di primario L1, che induce corrente nella bobina di secondario L2 e trovandosi in opposizione di fase, questa bobina L2 sottrae carica dalla base di Q1 portandolo nella zona di interdizione e quindi viene spento. Appena esaurita l'energia che provoca lo spegnimento di Q1, il transistor si riaccende nuovamente e si viene quindi ad instaurare una oscillazione fra le due bobine L1 e L2 ad una frequenza dell'ordine del MHz, che dipende dalla polarizzazione del circuito, dal numero di spire, dalle distanze e dai parametri parassiti, in particolare della capacità parassita dell'avvolgimento L2, che si creano tra spira e spira e con i piani di massa circostanti.

Il transistor Q1 selezionato come oscillatore, ha caratteristiche molto particolari: il parametro più

importante è f_t (prodotto Banda - Guadagno) che deve essere sufficientemente alto, così come la tensione di rottura delle giunzioni BE e BC ed anche il guadagno in DC h_{fe} , per poter garantire che il transistor riesca ad oscillare a frequenze dell'ordine del MHz e sostenere correnti e tensioni abbastanza elevate. Il transistor lavora in una zona altamente dissipativa e deve essere quindi dotato di dissipatore.

È stato inoltre previsto un piccolo stadio di modulazione AUDIO, realizzato tramite il MOSFET Q2, che agisce da resistore variabile in serie alla corrente di eccitazione della bobina regolata da Q1, per permettere anche l'iniezione di un segnale audio modulante in ingresso accoppiato tramite il condensatore C1 ed un partitore resistivo formato dalle resistenze R2 e R3, per polarizzare in conduzione il MOSFET Q2.

Il diodo D1 serve ad impedire un passaggio di corrente dal secondario della bobina verso l'alimentazione. Il LED LD1 è collegato in modo che si accenda quando l'avvolgimento di secondario richiama corrente dalla base portando il potenziale della base ad un valore negativo e quindi imposti una tensione pari alla sua soglia di conduzione e nel frattempo dia indicazione visiva del corretto funzionamento del circuito.

LA MODULAZIONE AUDIO

La bobina può anche "suonare", grazie all'effetto di compressione dell'aria prodotto dalle scariche elettriche. Per ottenere questo è necessario fornire un segnale audio in ingresso al circuito stampato, che abbia un'ampiezza minima di 10 volt picco-picco, in quanto un segnale di livello LINE (poche centinaia di millivolt efficaci, per intendersi) non sarebbe sufficiente a questo scopo e grazie al MOSFET Q1 la corrente elettrica che circola nella bobina verrà modulata in ampiezza.

Ovviamente vengono tagliate molte frequenze audio, ma si riuscirà comunque a sentire la musica prodotta dalle scariche. Questo principio è alla base dei "plasma-tweeter" commerciali, che sono dei riproduttori sonori progettati per diffondere la musica tramite la creazione di plasma.

PROGETTO DEL PCB

Prima di creare il circuito stampato è stato sperimentato il principio di funzionamento realizzando un avvolgimento di secondario a mano incollato su nastro biadesivo su una lastra di vetronite come mostrato in **Fig. 1**, utilizzando un filo di rame smaltato di sezione 0,315mm; in questo modo è stato

Fig. 1
Primo prototipo
dell'avvolgimento
di secondario
realizzato a mano.

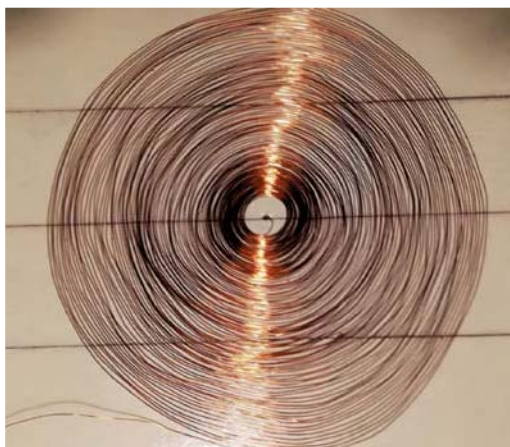




Fig. 2
Prototipo
PCB a rame
esposto
150mm x
150mm e
distanza tra
spira e spira
0,15 mm

possibile capire la sensibilità del circuito elettronico alla variazione dei parametri parassiti, collegati alle distanza spira-spira e con il piano di massa. Durante la progettazione del PCB sono stati provati diversi tipi di avvolgimento di secondario e di primario realizzati su circuito stampato, perché la vera criticità di questo progetto è la creazione di una spirale realizzata con le piste del circuito stampato stesso, compatta, con molte spire ravvi-

ciate, nel rispetto delle regole di sbroglio imposte dai vari costruttori di circuiti stampati. I primi tentativi di realizzazione del PCB, come mostrato in **Fig. 2**, sono stati fatti lasciando il rame dell'avvolgimento di primario e secondario esposto (senza "solder mask"), perché si temeva una forte alterazione dei parametri elettrici parassiti funzionali alla bobina di Tesla da parte del layer di "solder mask"; questo tipo di approccio, sebbene sia

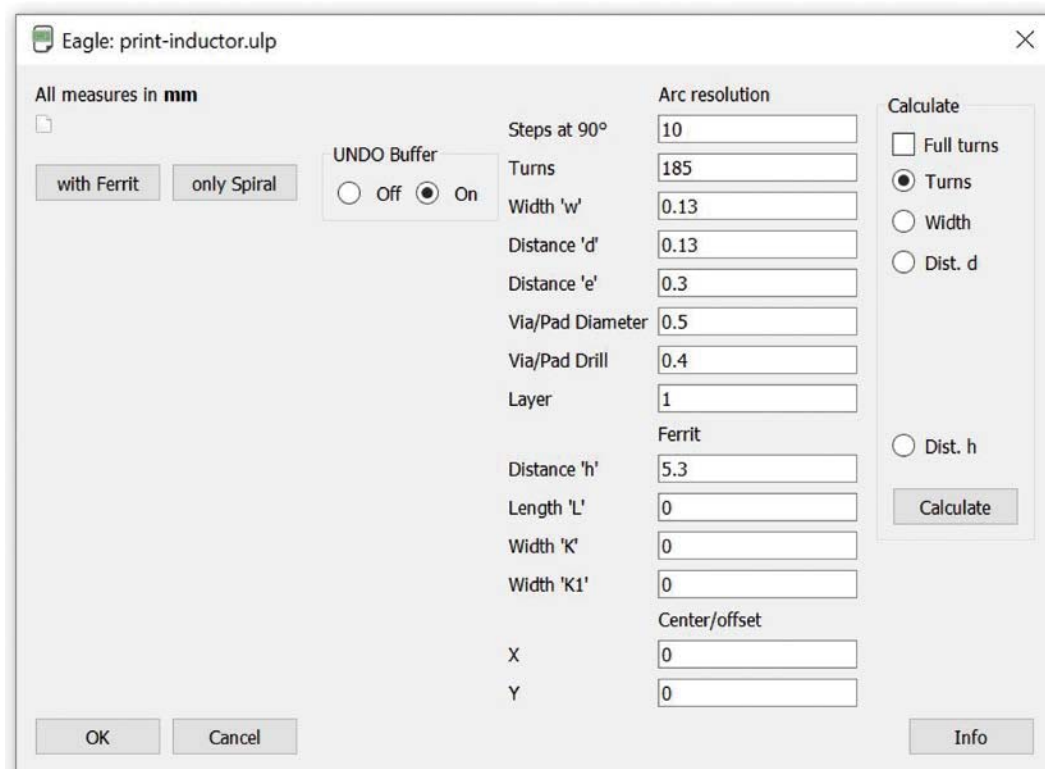
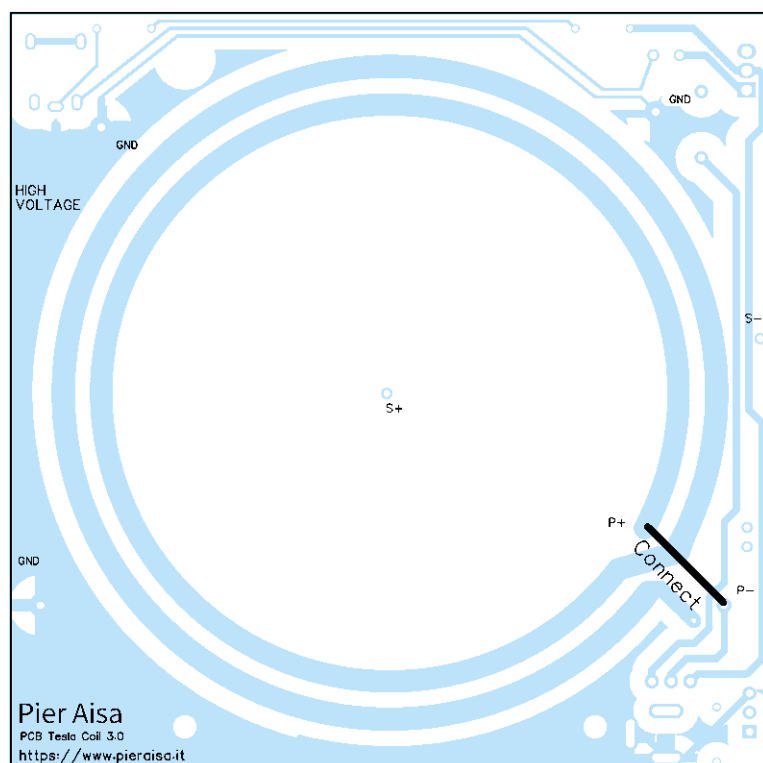


Fig. 3
Eagle script
"print-
inductor.ulp".

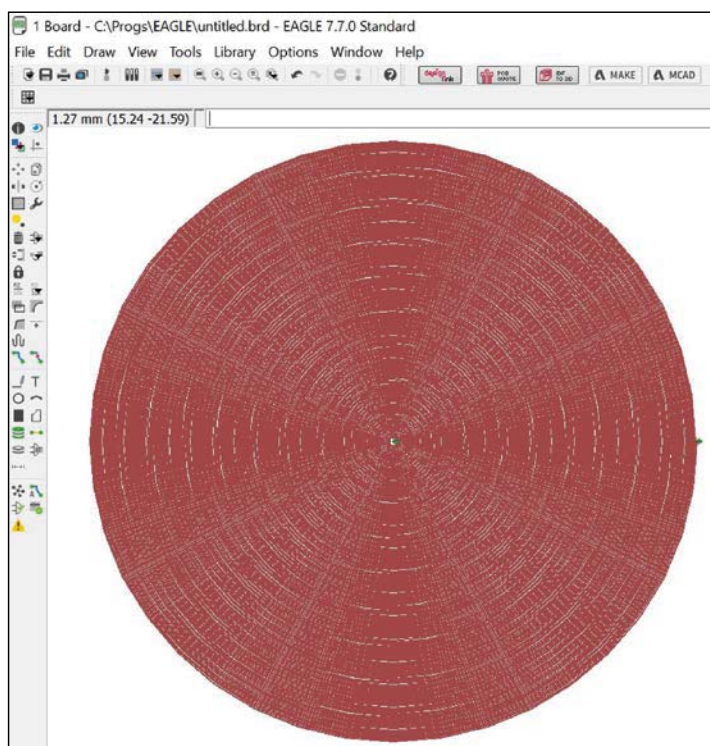


Elenco Componenti:

R1, R2: 82,5 Kohm
R3: 10 kohm
C1: 2,2 uF 63V elettrolitico
D1: MUR120
Q1: BD911 (transistor)
Q2: IRF3205PBF (mosfet)
LD1: led giallo 3 mm
L1, L2: bobina (vedi testo)

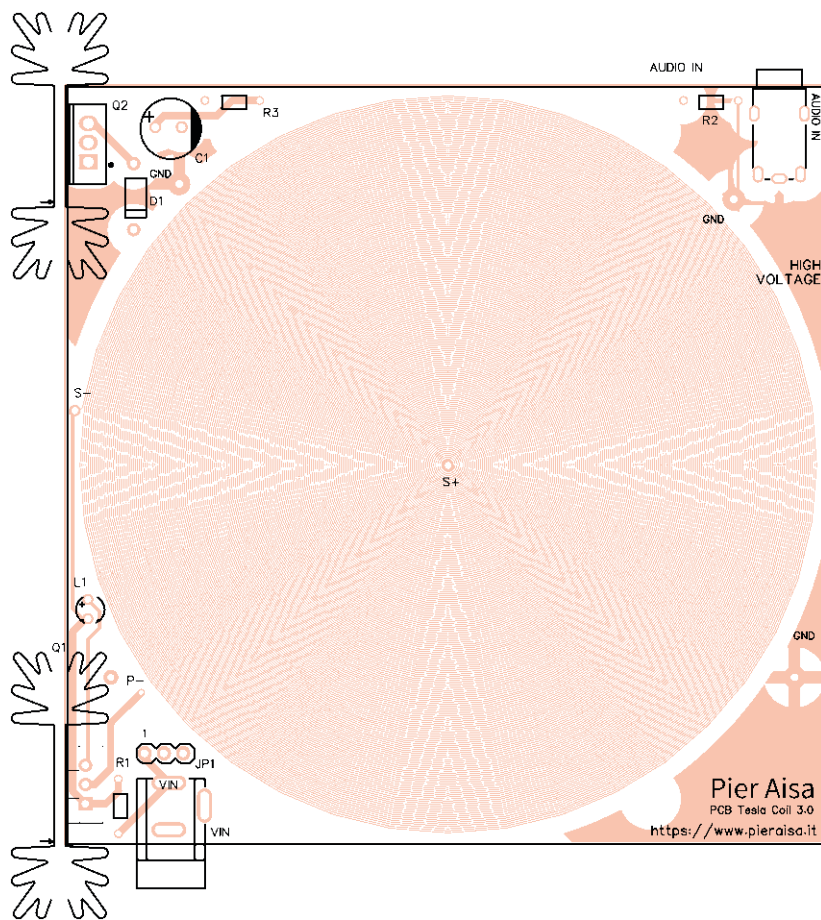
Varie:

- plug alimentazione
- jack femmina
- strip maschio 3 poli
- dissipatore 36x38x20
- stampato S1563 100x100



compatibile con il funzionamento della bobina, ha però presentato delle criticità realizzative da parte del costruttore con alcuni difetti di produzione, come bave ed ammanco di rame sulle piste sottili e circolari. Inoltre, il fatto di non avere la "solder mask", espone la bobina stessa a rischi di rottura dell'avvolgimento: infatti è molto delicata ed anche una sola goccia di stagno caduta per errore, potrebbe comprometterne la sua integrità. Inizialmente, si era tenuta una distanza tra spira e spira di 0,15mm e con questa regola per la realizzazione di 240 spire il circuito stampato aveva assunto dimensioni 150mm x 150 mm. La seconda e definitiva versione del circuito stampato è stata realizzata con l'obiettivo di ottenere un PCB con dimensioni massime di 100mm x 100mm, al fine di poter rientrare in un

Fig. 4
Avvolgimento a spirale in Eagle.



processo produttivo di tipo "prototipo a basso costo". Difatti questa dimensione di PCB è considerata da molti costruttori di circuiti stampati, come la taglia massima, entro la quale il costo è minimizzato, poiché considerato prototipale e calcolato ad area equivalente.

Fornendo questo tipo di vincolo si è realizzato l'avvolgimento di secondario massimizzando il numero di spire, nel rispetto delle regole di clearance e creepage imposte dal costruttore di PCB. Questa strategia è stata scelta perché risulta preferibile realizzare delle bobine più piccole, più compatte e a basso costo, che possono essere messe in serie per ottenere più spire al secondario, piuttosto che un'unica bobina con più spire. Per la realizzazione di un avvolgimento a forma di spirale è stato utilizzato una script di Eagle, chiamato "print-inductor.ulp"; lo si può lanciare usando il comando RUN ULP, dal menù File di Eagle. E viene richiesta l'immissione dei parametri costruttivi, che nel nostro caso sono riportati nella finestra di dialogo proposta in **Fig. 3**; i rispettivi valori sono stati calcolati per minimizzare l'area occupata.

Per sfruttare al massimo l'area assegnata, di

100mm x 100mm, è stata scelta una distanza fra spira e spira e una larghezza pista di 0,13mm, che rispetta la tipica regola di "minima distanza tra pista e pista" imposta dai costruttori di PCB pari a 5 mils (1 mil equivale a 1/1.000 di pollice), ossia 0,127mm. In questo modo sono state realizzate

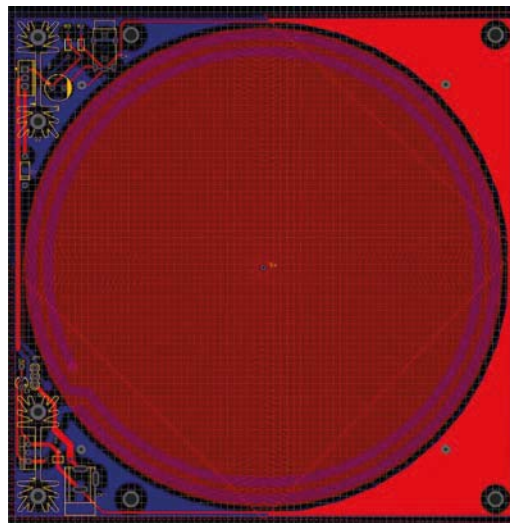


Fig. 5
Sbroglio PCB
in EasyEDA:
lato TOP
(rosso), lato
BOTTOM
(blu).

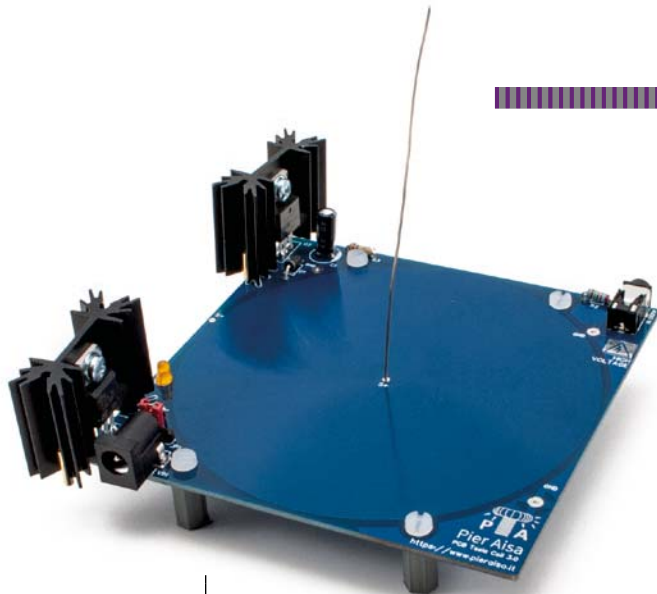


Fig. 6
Singola
PCBTC con
dissipatori
in verticale,
break-out
point e
distanziali
isolanti.

185 spire. Poi il file con estensione *.brd* prodotto da Eagle è stato importato in EasyEDA (ambiente di sbroglio PCB gratuito on-line). Per poter rispettare la massima dimensione assegnata è stato necessario modificare le geometrie dei dissipatori rimuovendo i fori metallizzati e facendo in modo che i transistor Q1 e Q2 abbiano il dorso del package a filo del circuito stampato. L'avvolgimento di primario è stato realizzato sul lato saldature (BOTTOM) con 2 spire, utilizzando una pista di grande superficie, per poter sostenere elevate correnti impulsive e per indurre un sufficiente campo magnetico; a questo proposito è molto importante anche la posizione dell'avvolgimento di primario, in modo che si trovi in

corrispondenza della parte esterna dell'avvolgimento secondario, come mostrato in **Fig. 5** dove si osserva in rosso (piste lato TOP) l'avvolgimento secondario ed in blu (piste lato BOTTOM) l'avvolgimento di primario e la costruzione dei piani di massa, necessari alla risonanza della bobina. La frequenza di risonanza dipende da molti fattori tra cui: il numero di spire, le capacità parassite del circuito stampato, la presenza di oggetti ferromagnetici nelle vicinanze e la polarizzazione del transistor Q1. La formula da utilizzare per il calcolo della frequenza di risonanza è quella classica che si utilizza per un circuito risonante LC e quindi all'aumentare del numero di spire ed all'aumentare della capacità parassita si abbassa la frequenza di risonanza.

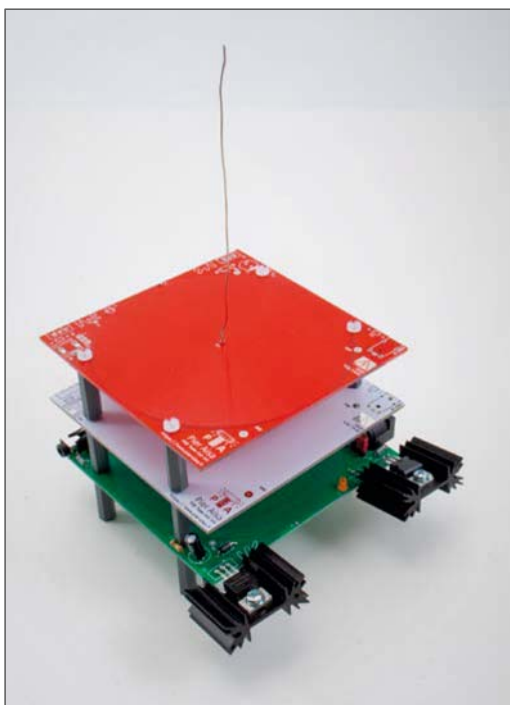
$$f_R = 1 / (2\pi \sqrt{LC}) = 1,31 \text{ MHz}$$

Abbiamo così ottenuto un PCB con un avvolgimento primario composto da 2 spire realizzate con una pista di larghezza 3mm e un avvolgimento secondario formato da 185 spire realizzate con una pista di larghezza 0,13 mm, resistenza di 121 ohm, induttanza pari a 1,194 mH, capacità parassita del secondario di 12,4 pF.

Come si vede nelle foto del prototipo e nel piano di montaggio, il circuito stampato fa anche uso di due piani di massa, uno sul lato componenti uno

Fig. 7
Montaggio dei
dissipatori in
orizzontale.

Fig. 8
Collegamento
in serie con filo
metallico.



sul lato saldature, che aiutano a mantenere stabili i parametri di capacità parassita, che concorrono alla oscillazione della bobina ed ha dimostrato un ottimo fattore di accoppiamento tra primario e secondario: infatti già ad una tensione di alimentazione pari a 5 volt si innesca l'oscillazione risonante alla frequenza calcolata.

REALIZZAZIONE PRATICA

Passiamo adesso alla fase costruttiva, partendo dal circuito stampato, per la cui realizzazione potete scaricare dal nostro sito www.elettronica.in le tracce lato rame. Ricordiamo che questo nostro progetto nasce modulare, nel senso che è possibile realizzare più circuiti e impilarli per ottenere generatori con differenti caratteristiche, ovvero delle scariche di maggiore lunghezza. Vedremo questo aspetto tra breve.

Restando sul circuito, una volta inciso e forato lo stampato, si parte con il montaggio dei componenti del PCBTC, che è molto semplice perché ci sono pochi componenti da saldare e montare, seguendo le indicazioni del piano di montaggio.

È necessario rispettare le indicazioni di polarità riportate sul circuito stampato per i diodi, i transistor ed il condensatore. Il circuito stampato deve essere sollevato dal piano di appoggio tramite torrette e viti in materiale isolante, per evitare di richiudere delle scariche all'interno del PCB stesso.

Bisogna inoltre realizzare un ponticello di collegamento fra i due PAD della serigrafia siglati "Connect" (riferitevi al piano di montaggio), utilizzando un conduttore metallico di diametro 0,5mm, ovvero uno spezzone avanzato da un terminale di componente (tipo una resistenza).

I transistor Q1 e Q2 ed i rispettivi dissipatori KK1 e KK2, vanno montati in verticale se intendete realizzare un generatore a singolo circuito (**Fig. 6**) ovvero in orizzontale, laddove intendiate realizzare l'assemblaggio di più PCBTC impilati (**Fig. 7 e Fig. 8**).

Per creare un punto preferenziale di emissione delle scariche è necessario creare un Break-out point, ossia un punto in cui il campo elettrico sia superiore al valore presente in altri punti della ba-setta, tanto che da esso si possa liberare la scarica elettrica in aria. La fine dell'avvolgimento secondario (PAD con serigrafia S+) quindi viene collegato ad un filo metallico, che nella sua punta funziona da emettitore preferenziale di scariche.

Per realizzare il Break-out point dovete utilizzare un filo di rame rigido stagnato, della lunghezza di 8 cm, saldato nel terminale centrale della bobina, contrassegnato dalla sigla S+ sul PAD: lo vedete

AVVERTENZE E PRECAUZIONI



La bobina PCBTC quando è in funzione dà luogo alla formazione di effetto corona nelle zone ad alta tensione. Gli archi elettrici sono in grado di ionizzare l'aria in modo da produrre una piccola quantità di ioni negativi ed ozono (O₃) e quindi è necessario areare il locale frequentemente, per evitare l'inalazione di Ozono, che risulta nocivo alla salute. Inoltre, data la presenza di un forte campo elettrico si sconsiglia l'utilizzo di questa bobina a tutti coloro che indossano dei dispositivi medici sensibili ai campi elettrici e magnetici come pacemakers, perché la bobina potrebbe alterarne il regolare funzionamento. Si consiglia di posizionare la bobina lontano da dispositivi elettronici, per evitare sollecitazioni troppo forti a causa dei campi elettromagnetici intensi, dovuti alla oscillazione della bobina ad una frequenza di qualche MHz, che è una radiofrequenza che si propaga molto bene; si potrebbero infatti, notare delle perturbazioni negli oggetti nei dintorni quali schermi LCD e televisori.

Questa bobina è stata chiamata "la bobina di Tesla per tutti", proprio per la sua semplicità di realizzazione e la bassa pericolosità infatti di solito le bobine Tesla non hanno questi presupposti, perché solitamente implicano altissime tensioni e correnti pericolose, ma non bisogna comunque sottovalutare il rischio elettrico. Se si dovesse venire a contatto con la scarica, si avrebbe comunque la certezza che questa non possa penetrare negli organi interni vista la sua elevata frequenza e l'effetto pelle. Non bisogna toccare le scariche a mani nude perché provocano delle bruciature.

bene nella **Fig. 6**. Notate che è possibile provare anche altre lunghezze per questo filo rigido, ma la cosa importante è che sia verticale ed abbastanza lontano dal piano del PCB, in modo da evitare scariche elettriche dirette sulle spire, che sicuramente non gioverebbero.

Nel caso si vogliano collegare più PCB in cascata (cioè impilati) è necessario collegare la parte finale del filo di Break-out point del primo PCB nella PAD contrassegnata con S- (inizio avvolgimento secondario) del secondo PCB come mostrato nella **Fig. 8**. Gli ulteriori PCB dovranno essere distanziati con torrette e viti, rigorosamente in materiale isolante (poliestere).

Quando una elevata differenza di potenziale, superiore a diverse migliaia di volt viene applicata tra due elettrodi immersi in un mezzo gassoso, si innesca il fenomeno della "ionizzazione a valanga", che implica l'accelerazione delle particelle, che urtando con le molecole d'aria produce una grande quantità di cariche libere nell'aria. Si viene così a creare attorno agli elettrodi una nuvola di cariche che generano una corrente elettrica diretta visibile ad occhio nudo con una luce violacea e con un crepitio, chiamata "effetto corona".

La corona negativa appare di dimensioni maggiori rispetto a quella positiva poiché gli elettroni possono allontanarsi maggiormente dalla zona di ionizzazione e così l'aria ionizzata può estendersi oltre. L'effetto corona è un ottimo generatore di ozono O₃. La corona negativa ne genera in quantità molto maggiore di quella positiva. Dopo un po' di secondi di funzionamento del MiniHV si potrà notare un odore pungente, come avviene durante un forte temporale, dovuto appunto alla produzione di molecole di ozono.

lazione audio. È necessario alimentare il circuito con una alimentazione a +24Vcc ed almeno 30W di potenza disponibile. Se la bobina è in funzione regolarmente si deve accendere il LED L1. Avvicinare una lampada a tubo a gas alla bobina PCBTC, senza toccarla (toccando solo il vetro, che risulta un ottimo isolante) e verificare che la lampada si illumini (**Fig.9**). Avvicinare la punta di un cacciavite, stando attenti a non toccare con le mani la parte metallica del cacciavite, al break-out point e verificare che si inneschi l'arco elettrico (**Fig.10**). Nella versione a 2 o 3 PCB, considerata l'energia in gioco, la bobina non può funzionare in modalità continua, non bisogna superare un tempo di funzionamento di 15 secondi ed è necessario attendere circa 1 minuto per la riaccensione.

PRIMA ACCENSIONE ED UTILIZZO

Una volta ultimato il montaggio inserire un jumper in posizione 1-2 del pin header con la sigla JP1, se si desidera il solo funzionamento in scarica o in posizione 2-3 per il funzionamento con modu-

TEST E MISURE

Nel nostro laboratorio sono state fatte diverse prove sui prototipi variando la tensione di alimentazione da 12 volt a 30 volt. La tensione di 24 volt è risultata il miglior compromesso fra corrente assorbita e intensità del campo elettrico genera-

Fig. 9
Accensione di una lampada a tubo gas tramite il campo elettrico generato.

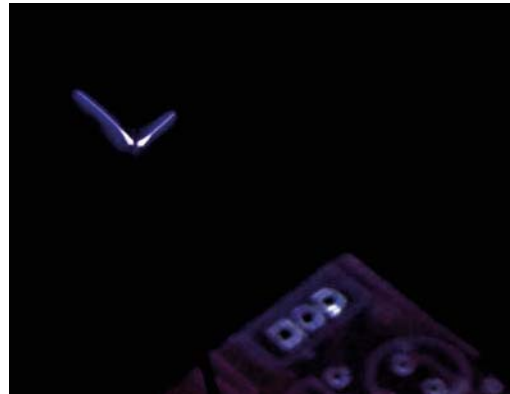
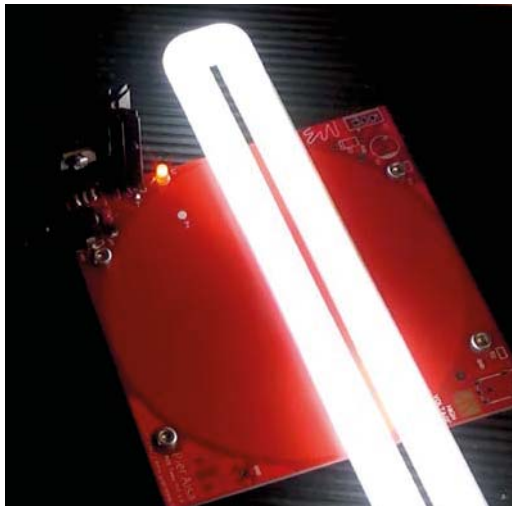


Fig. 10
Scariche in uscita dal break-out point.

Tabella 1
Risultati osservati con le diverse configurazioni delle bobine.

PCB IN SERIE	NUMERO SPIRE	RESISTENZA DI SECONDARIO (ohm)	CORRENTE ASSORBITA (A)	FREQUENZA (MHz)	LUNGHEZZA ARCO (mm)
1	185	121	1,01	1.311	7
2	370	242	1,03	1.033	13
3	555	359	1.16	0.875	15

to; infatti in corrispondenza di questa tensione la scarica si innesca in maniera naturale, senza dover avvicinare punte metalliche alla fine dell'avvolgimento secondario realizzato con un filo metallico chiamato Break-out point, perché per il "principio delle punte" in questa zona il campo elettrico è più intenso ed in grado di rompere il dielettrico dell'aria. Nella **Tabella 1** sono stati riassunti i parametri misurati sul prototipo del Tesla-coil utilizzando una tensione di alimentazione di 24V su diverse configurazioni, con uno o più PCB collegati in serie e come si vede la lunghezza dell'arco innescato aumenta, con una proporzione che non è lineare, ma oltre ad un certo numero di spire rimane praticamente la stessa.

La **Fig.10** mostra la singola bobina in funzione ed in questo caso la scarica presenta due punte della lunghezza di circa 7 mm; se si avvicina un oggetto metallico al Break-out Point, si crea un arco elettrico che può raggiungere una lunghezza fino a 15 mm.

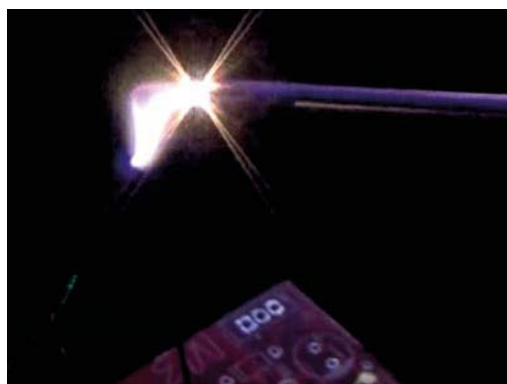
La bobina può funzionare con tensioni massime fino a 30 volt; oltre questa tensione invece potrebbero innescarsi archi elettrici tra l'avvolgimento secondario e le altre parti del PCB. Per evitare questi archi indesiderati nella versione 3 di PCBTC, sono state rimosse tutte le parti metalliche sulle pad e si richiede l'utilizzo di torrette in materiale plastico.

La **Fig. 11** vi mostra un punto in cui è avvenuta una scarica indesiderata a causa dell'utilizzo di torrette metalliche e PAD metalliche della versione 2 di PCBTC. La **Fig. 12** e la **Fig. 13** propongono l'effetto causato dall'avvicinamento della punta di un cacciaviti tenuto in mano: l'arco scaturisce dal fatto che l'isolante del cacciaviti non è in grado di resistere a tale campo elettrico, quindi una corrente pur piccola lo attraversa e passa dal nostro corpo; è quanta ne basta a chiudere il circuito, ma non preoccupatevi, perché è innocua.

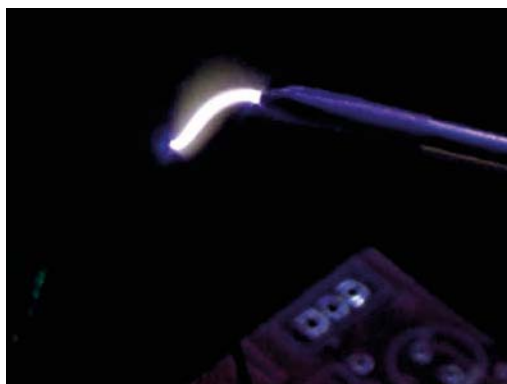
Considerato che viene prodotto un forte campo elettrico questa bobina può essere utilizzata anche per verificare il corretto funzionamento di elementi sensibili al campo elettrico in maniera facile ed efficace, come ad esempio i LED o lampade a tubo a scarica di gas che riescono ad essere facilmente accese perché grazie al campo elettrico avviene la ionizzazione del gas contenuto. I LED sensibili al campo elettrico del Tesla-coil sono tipicamente quelli a luce bianca, giacché sono LED a luce blu o ultravioletta il cui interno è rivestito di fosfori che stimolati da tale luce restituiscono fotoni di luce bianca; tali fosfori possono essere ugualmente eccitati dal campo ad alta tensione. □



← **Fig. 11**
Punto di scarica indesiderato.



← **Fig. 12**
Innesco dell'arco elettrico con il cacciavite.



← **Fig. 13**
L'arco elettrico.



Cosa occorre?

I componenti utilizzati in questo progetto sono di facile reperibilità. Il PCB può essere realizzato in diversi colori utilizzando il servizio di stampa circuiti stampati (cod. PCBPRODUCTION) direttamente dal sito www.futurashop.it.

I prezzi si intendono IVA compresa.

Il materiale va richiesto a:

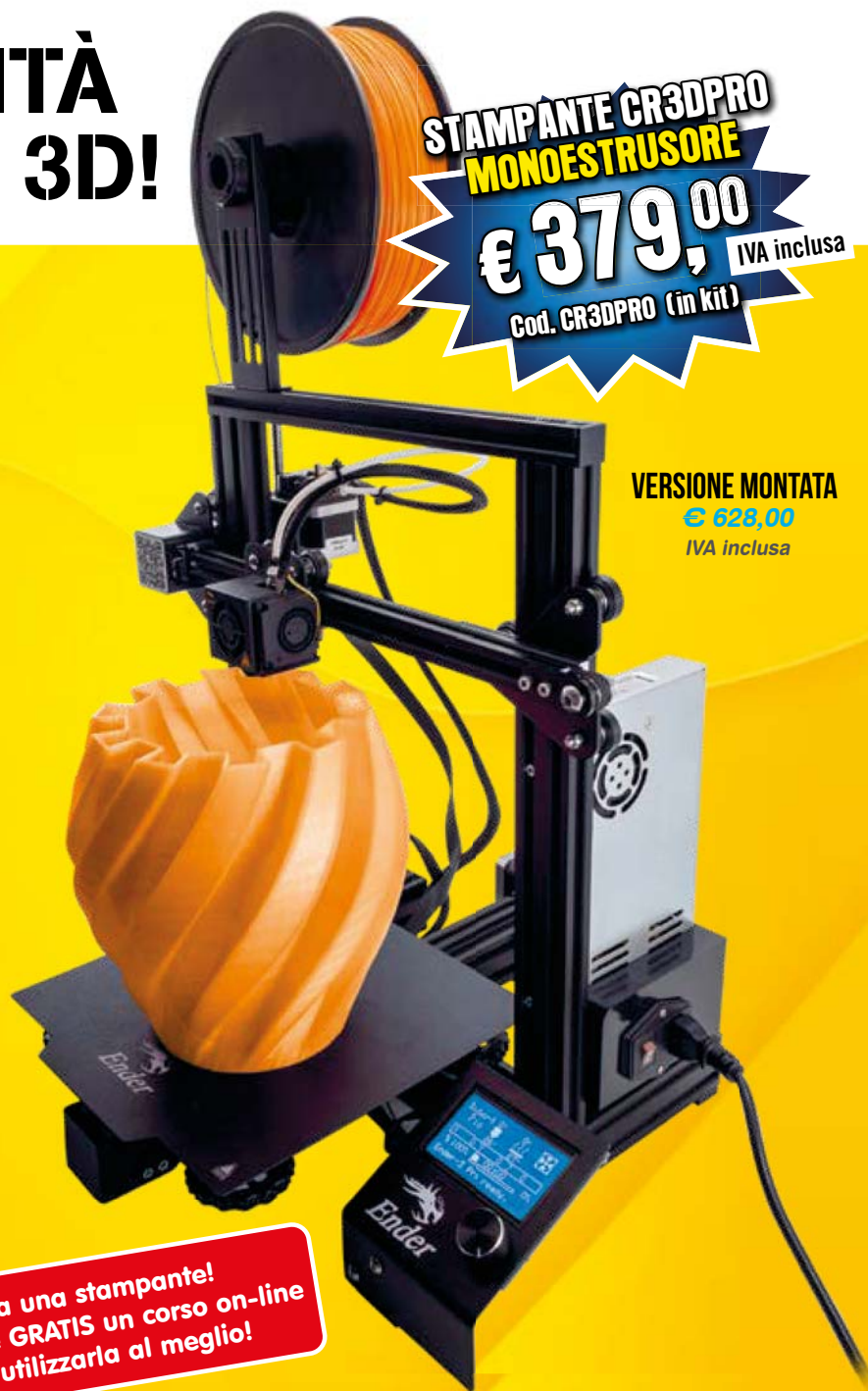
Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 - <http://www.futurashop.it>

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



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

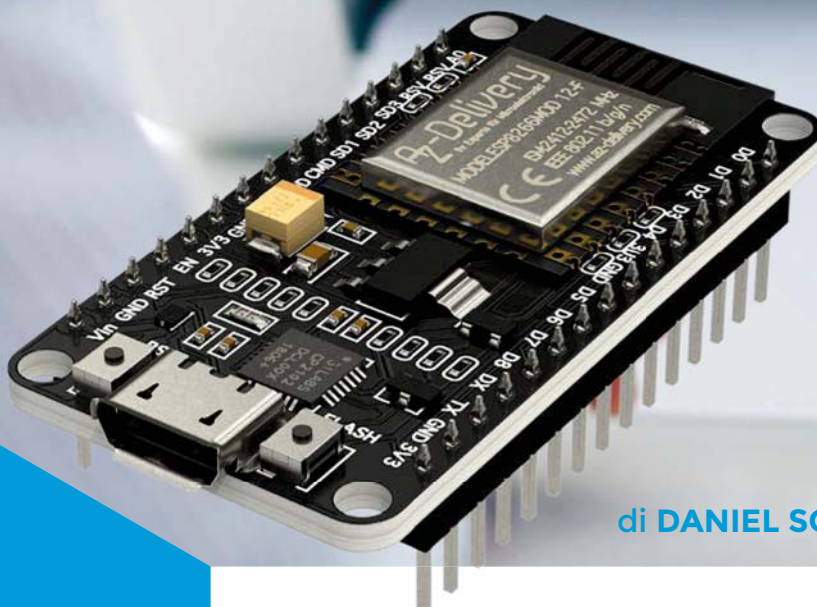
CRCP3IN1

**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

STAMPIAMO ON-LINE TRAMITE SMARTPHONE



di DANIEL SCAINI

Controlliamo un dispositivo seriale con il modulo WiFi ESP8266. I comandi saranno inviati da remoto attraverso il protocollo MQTT.

Il Cloud ha cambiato il modo di fruire varie tecnologie e consentito applicazioni un tempo limitate alle reti locali o alle VPN; ma ha anche creato nuove possibilità come il Cloud Printing, ossia la fruizione di una stampante affacciata al web come servizio, che permette di utilizzarla da qualsiasi dispositivo anche senza installare driver o appartenere alla stessa rete. Ciò rende possibile inviare documenti in stampa in modo semplice e sicuro, anche se siamo in viaggio o fuori casa e senza la necessità di installare alcun software. Da non sottovalutare anche la semplicità di questo sistema, che sfruttando le reti già presenti, senza il peso di nuove installazioni hardware, azzerà le distanze e rende fruibile la risorsa ovunque. Tutto ciò è ottenuto con una semplice comunicazione publisher/subscriber attraverso il protocollo MQTT, secondo il



Fig. 1
Stampante
P3 connessa
in seriale con
il NodeMCU
ESP8266.

quale un client o publisher (il device da cui si vuole stampare) invia il documento ed un subscriber (stampante) che riceve la risorsa; il tutto in tempo reale. A differenza di una comunicazione Client-Server, questa è molto più precisa e richiede meno configurazioni, giacché non sono necessarie, per la parte client, aperture di porte o modifiche di alcun genere.

Rispetto a una comunicazione web più classica, qui non vi è polling, ovvero continua ispezione da parte del client, il quale ricerca se vi è o meno una risorsa a lui destinata; infatti la comunicazione è basata su una connessione permanente all'interno della quale vengono "pushati" (inviati) i dati, una volta pronti.

In queste pagine spiegheremo l'architettura distribuita che rende possibile la stampa remotizzata e lo faremo con un esempio applicativo: la stampa dell'ordinazione al ristorante eseguita da uno smartphone direttamente su una stampante Custom P3 da cui il personale prenderà la "comanda". L'interfacciamento della stampante, che è seriale, avviene tramite un NodeMCU (**Fig. 1**).

Attraverso la stessa tecnica è possibile remotizzare ogni servizio e renderlo fruibile da ogni device, sia esso una stampante, uno scanner o semplicemente una lampadina.

LINGUAGGIO DI PROGRAMMAZIONE

Viste la modularità e la distribuzione del siste-

ma, utilizzeremo due linguaggi: il C++, non quello standard chiaramente, ma un "dialetto" da esso derivato, progettato per poter agire su piattaforme come Arduino e similari. Per supportare lo sviluppo su questi tipi di architettura e con questo linguaggio c'è a disposizione la piattaforma, derivata dall'IDE creato per il linguaggio di programmazione Processing, ovvero Arduino IDE.

Questo ambiente è molto diffuso e perciò integra una serie di librerie e moduli in grado di facilitare di gran lunga il nostro lavoro. Inoltre è possibile aggiungere librerie open source esterne.

Il secondo linguaggio utilizzato è l'HTML, ossia una parte di HTML puro per scrivere la pagina web, una parte di CSS per poterla ornare e rendere più accattivante ed infine una parte di Javascript responsabile della comunicazione e della creazione della logica di base dell'applicazione.

Per quest'ultima andremo ad importare una libreria che ci semplificherà di molto il lavoro di interazione con la UI: JQuery. Si tratta di una libreria open source che nasce con l'obiettivo di semplificare la selezione, la manipolazione e la gestione di elementi DOM in pagine HTML; infatti il framework rende disponibile una serie di metodi per gestire al meglio aspetti grafici e strutturali riuscendo a mantenere compatibilità fra i vari browser.

HARDWARE

Come già accennato precedentemente durante l'introduzione, l'hardware su cui baseremo tutto il nostro Progetto è il chip ESP8266 (**Fig. 2**). Nel dettaglio utilizzeremo la demo board NodeMCU V1 sulla quale è montato il chip ESP8266 della Espressif System, la quale deve la sua fortuna e la sua rapida espansione all'introduzione e al porting all'interno del suo core della libreria MQTT di cui parleremo più avanti. Questa piattaforma, grazie alle sue ridottissime dimensioni, è l'ideale per la prototipazione di applicazioni IoT (Internet of Things) perchè al suo interno ha un modulo WiFi compatibile con lo standard 802.11 b/g/n a 2,4 GHz, dispone di stack TCP/IP integrato, potenza di uscita di 19,5 dBm, interfaccia dati (UART/HSPI/I²C/I²S, IR, GPIO/PWM) e antenna su PCB. È governata da un microcontrollore Tensilica 32-bit RISC CPU Xtensa LX106, con un clock di 80MHz, al quale si affiancano 4MB di flash e 64K di SRAM. A differenza di Arduino non possiede una vera e propria EEPROM, ma questa viene simulata alla perfezione sfruttando la Flash, rendendolo in grado di memorizzare alcune impostazioni del programma o dati esterni. Lavora ad una tensione di

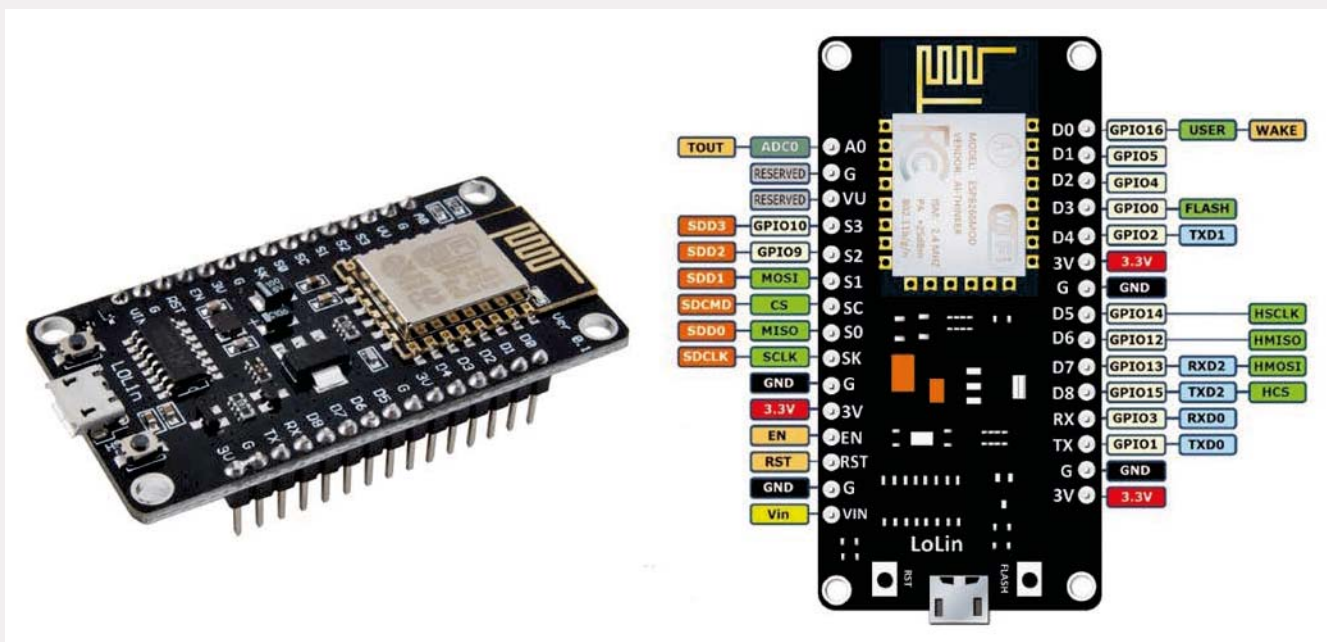


Fig. 2
NodeMCU con
ESP8266.

3,3V ma in ingresso accetta una tensione che può variare da 3 a 12V. Mette a disposizione, oltre che ad una microUSB per poterlo programmare agevolmente, ben 16 pin digitali (DIO), un pin analogico (ADC), un pin per la SPI ed un pin per I²C. La stampante P3 utilizzata nel nostro progetto è una termica POS creata per i settori Retail e Hospitality, dotata di connessione ethernet, USB e seriale; utilizzeremo quest'ultima inserendole un connettore DB-9 da cui preleveremo i fili come in **Fig. 3** per collegarla al modulo ESP secondo la **Tabella 1**. Il pin RXD (D5) non verrà usato.

MQTT

Message Queue Telemetry Transport, è un protocollo di messaggistica, molto leggero, di tipo publish-subscribe ed è utilizzato laddove vi è una banda limitata o si vuole creare poco impatto a livello di rete. Creato da Andy Stanford-Clark e Arlen Nipper nel 1999, è stato adattato in primis per monitorare gli oleodotti. L'obiettivo era quello di creare un protocollo veramente leggero, efficiente in termini di lunghezza di banda e che consumasse poca batteria, argomento spinoso sia allora che adesso. Questo protocollo vede però una nuova vita nel 2013 quando viene rilasciata da IBM la versione 3.1 e comincia ad affermarsi grazie anche ad una forte spinta del mercato verso l'IoT (Internet of Things). Uno degli esempi di maggiore risalto dell'uso dell'MQTT è sicuramente Facebook Messenger, dove

sarebbe stato adottato per migliorare la messaggistica e limitare l'uso di banda e il consumo energetico.

HomeAssistant, una delle migliori piattaforme open source per gestire la domotica, oltre ad integrare numerosi servizi per poter interagire con le numerose periferiche che supportano il protocollo, ha rilasciato un plugin per riuscire lui stesso ad assumere la funzionalità di broker.

Ma andiamo a vedere nel dettaglio come funziona. In questo protocollo ci sono due tipi di attori: un broker e vari client. Quest'ultimo è un qualunque dispositivo, da un semplice microcontrollore fino ad un server completo, il quale esegue una libreria MQTT ed è connesso ad un broker. Questo attore invia al broker delle informazioni, chiamate payload, classificate in base al loro topic. Le informazioni infatti sono organizzate in base all'argomento, chiamato topic. Quando un client invia un nuovo payload ad un broker, questo rilancia l'informazione a tutti i client ad esso connessi su un determinato topic.

Il client non ha idea del numero o delle posizioni degli "abbonati" connessi a quel topic. La **Fig. 4** propone uno schema di come funziona. Analizziamo invece rapidamente il primo, il broker: questo è generalmente un software in esecuzione su un PC o una scheda embedded a microcontrollore e ci sono varie implementazioni sia open source che proprietarie. Questo funziona come da "ufficio postale" ed

Fig. 3
Connettore DB-9 RS232 con i fili per l'interfacciamento al NodeMCU.

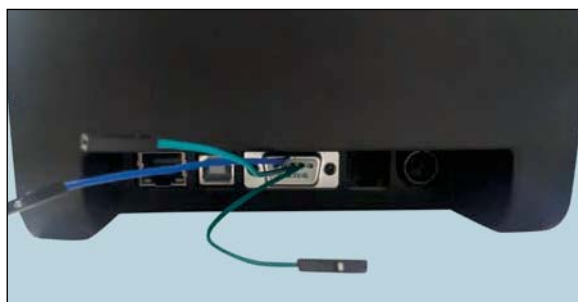


Tabella 1
Connessioni per il test.

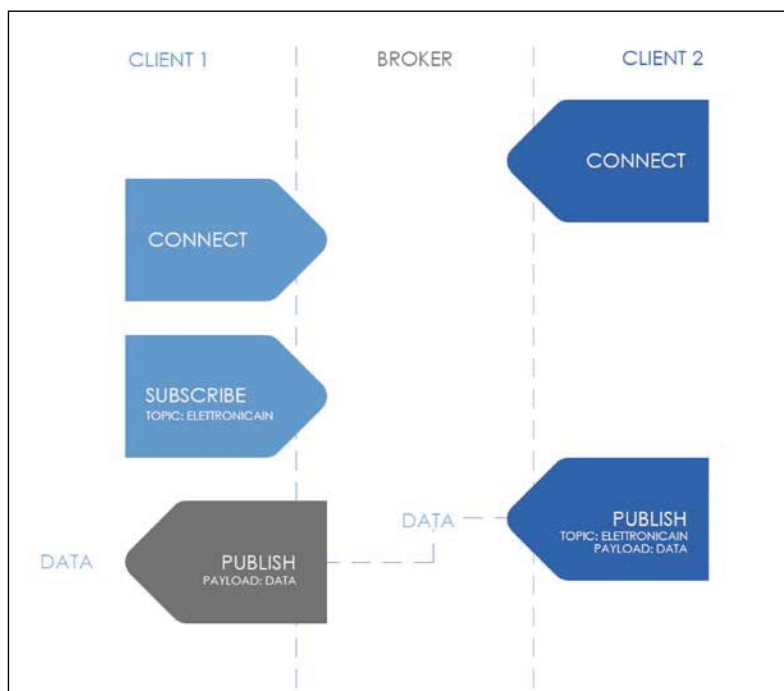
NODEMCU	DB-9
PIN 6 (D6)	3
PIN 5 (D5)	2
[Non usato nel nostro codice]	
PIN GND	5

è l'anello fra i vari client: il suo compito come accennato precedentemente è quello di rilanciare le informazioni in arrivo da un client a tutti i client connessi a quel determinato topic. Nel caso in cui non vi siano client connessi a quell'argomento, e il messaggio non è stato marchiato come da conservare, il payload viene scartato.

MQTT si basa su protocollo TCP per la trasmissione dei dati e solitamente coinvolge la porta 1883 nel caso di comunicazione in chiaro, oppure la porta 8883, in caso di comunicazione criptata. È stata sviluppata anche una variante chiamata MQTT-SN che viene utilizzata su altri mezzi di trasporto come UDP o Bluetooth.

A livello di sicurezza il broker può implementare una connessione sicuro basata su TLS.

Fig. 4
Struttura della comunicazione MQTT.



IL NOSTRO BROKER

Il broker è quindi una figura molto importante per l'architettura MQTT e di conseguenza, in un'applicazione è fondamentale scegliere quello giusto. Per consentire a chiunque di avvicinarsi al mondo MQTT ne esistono diversi free (**Tabella 2**) e tra questi EMQX (<https://www.emqx.io/mqtt/public-mqtt5-broker>) è il più completo. Offre in maniera chiara tutte le 4 vie per potersi connettere e cominciare a interagire con i vari client. Ovviamente i broker pubblici hanno il difetto di avere molto traffico e quindi bisogna usare topic sempre molto ricercati e unici per evitare di incorrere in false comunicazioni. Inoltre con essi non è possibile usare una vera e propria autenticazione. Per ovviare a questi inconvenienti è sempre consigliabile, dove possibile e per applicazioni non banali, crearsi il proprio broker.

Nel nostro caso consigliamo di utilizzare una Raspberry Pi, quindi per prima cosa, dal sito ufficiale www.raspberrypi.org scarichiamo ed installiamo Raspberry PI Imager, ovvero il tool responsabile di creare un sd card bootabile in grado di far partire il dispositivo. Ora procuriamoci il sistema operativo che in seguito installeremo a bordo della nostra piattaforma.

Quale OS scegliere? Una qualsiasi distribuzione server può andare bene, ma nel dettaglio scarichiamo la Ubuntu Server 20.04.2 LTS a 64bit (<https://ubuntu.com/download/raspberry-pi>). Una volta completato il download apriamo RPI Imager, inseriamo una SD card, e clicchiamo CHOOSE OS. 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 (**Fig. 5**).

Una volta completata la scrittura del nostro sistema operativo, siamo pronti per estrarre la scheda SD dal PC ed inserirla all'interno del nostro

Tabella 2
Confronto fra i vari broker MQTT pubblici.

NOME	INDIRIZZO	TCP	TLS	WS	WSS	MSG. RETENTION	PERSISTENT
Eclipse	mqtt.eclipse.org	1883	N/A	80	443	Si	Si
Mosquitto	test.mosquitto.org	1883	8883, 8884	80	N/A	Si	Si
HiveMQ	broker.hivemq.com	1883	N/A	8000	N/A	Si	Si
Flespi	mqtt.flespi.io	1883	8883	80	443	Si	Si
Dioty	mqtt.dioty.co	1883	8883	8080	8880	Si	Si
Fluux	mqtt.fluux.io	1883	8883	N/A	N/A	N/A	N/A
EMQX	broker.emqx.io	1883	8883	8083	8084	Si	Si

Raspberry, opportunamente collegato alla rete. Per connetterci a quest'ultimo, una volta acceso, utilizziamo PuTTY o qualunque programma permetta di lanciare una sessione SSH (**Fig. 6**). Al primo accesso il sistema chiederà utente e password, che per questo sistema operativo sono predefinite "ubuntu" per l'utente e "ubuntu" per la password. Ovviamente ci verrà chiesto di cambiare quest'ultima per ragioni di sicurezza.

Una volta completata la configurazione del OS passiamo alla configurazione del pacchetto del broker MQTT.

Come broker utilizzeremo mosquitto, un pacchetto che può essere installato anche attraverso apt ma così facendo perdiamo tutta la parte riguardante i web socket.

Andiamo quindi a scaricare e compilare le librerie necessarie per la futura compilazione del nostro broker, eseguendo gli step elencati e commentati nella **Tabella 3**.

Ora siamo pronti per scaricare i sorgenti del nostro broker: lo facciamo attraverso i comandi elencati e descritti nella **Tabella 4**.

Fatto ciò non ci resta che modificare la configurazione al fine di ottimizzare il nostro nuovo broker. Andiamo quindi ad aprire il config:

```
sudo nano /etc/mosquitto/mosquitto.conf
```

In basso aggiungiamo un'ulteriore riga:

```
include_dir /etc/mosquitto/conf.d
```

nel quale si fa riferimento ad una configurazione esterna, in modo tale da non dover agire su questo

file, molto complesso, ma su uno vuoto.

Anche qui per uscire e salvare le modifiche cliccare su "Ctrl+X" e poi "y". Verifichiamo che sia presente la cartella conf.d, altrimenti la creiamo attraverso:

```
sudo mkdir /etc/mosquitto/conf.d
```

Ora creiamo e modifichiamo la nostra configurazione vuota:

```
sudo nano /etc/mosquitto/conf.d/default.conf
```

Ed incolliamo le seguenti righe:

```
persistence true
persistence_location /var/lib/mosquitto/
allow_anonymous true
port 1883
listener 1884
protocol websockets
```

E come già fatto in precedenza usciamo e salviamo attraverso "Ctrl+X" e poi "y".

Ora avviamo mosquitto digitando:



Fig. 5
Scrittura della SD per Raspberry Pi.

Fig. 6
Sessione SSH di
puTTY.

```
ubuntu@ubuntu: ~
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Wed Apr  7 15:40:25 UTC 2021

System load:  0.0           Temperature:   44.8 C
Usage of /:   9.0% of 29.14GB Processes:      131
Memory usage: 5%           Users logged in: 0
Swap usage:   0%           IPv4 address for eth0: 192.168.1.130

* Introducing self-healing high availability clusters in MicroK8s.
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

  https://microk8s.io/high-availability

30 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Wed Apr  7 15:26:47 2021 from 192.168.1.129
ubuntu@ubuntu:~$
```

```
sudo mosquitto -c /etc/mosquitto/mosquitto.conf
```

Ora possiamo connetterci da un qualsiasi client nella stessa rete. Per testare la connessione consiglio un MQTTBox, un semplice tool per Windows, in grado di prevedere diversi settaggi ed un'ottima UI. Vediamo quindi come testare il broker: una volta

installato ed aperto il tool clicchiamo su Create MQTT Client ed inseriamo i dati come in **Fig. 7**, quindi clicchiamo su Save. In alto vi è un indicatore che segnala la buona riuscita o meno della connessione (**Fig. 8**) ed inoltre possiamo monitorare lo stato della connessione o i problemi dalla finestra SSH del broker ancora aperta (**Fig. 9**). A questa

Tabella 3
I comandi per scaricare e scompattare le librerie.

STEP	COMANDO	DESCRIZIONE
1	<pre>sudo apt-get update sudo apt-get install build-essential python quilt python-setuptools python3 sudo apt-get install libssl-dev sudo apt-get install cmake sudo apt-get install libc-ares-dev sudo apt-get install uuid-dev sudo apt-get install daemon sudo apt-get install libwebsockets-dev</pre>	Aggiornamento pacchetti Installazione libreria python
2	<pre>wget http://archive.ubuntu.com/ubuntu/pool/ universe/c/cjson/cjson_1.7.14.orig.tar.gz</pre>	Scarichiamo la libreria cJSON per riuscire a trattare in modo corretto le strutture dati
3	<pre>tar zxvf cjson_1.7.14.orig.tar.gz</pre>	Scompattiamo il tutto
4	<pre>cd cJSON-1.7.14</pre>	Rechiamoci nella cartella in questione
5	<pre>cmake .</pre>	Configuriamo il tutto per poter essere compilato
6	<pre>make</pre>	Compiliamo la libreria
7	<pre>sudo make install</pre>	Installiamo la libreria



Tabella 4
Comandi per scaricare i sorgenti del broker MQTT.

STEP	COMANDO	DESCRIZIONE
1	wget https://mosquitto.org/files/source/mosquitto-2.0.4.tar.gz	Scarichiamo i sorgenti del nostro broker
2	tar zxvf mosquitto-2.0.4.tar.gz	Scompattiamoli
3	cd mosquitto-2.0.4	Rechiamoci nella cartella in questione
4	nano config.mk	Apriamo il file per poter modificare la configurazione
5	WITH_WEBSOCKETS = no in WITH_WEBSOCKETS = yes	Editiamo questa riga trasformandola come segue
6	Ctrl-X e Y successivamente	Per salvare ed uscire
7	make	Avviamo il processo di compilazione
8	sudo make install	Installiamo il broker
9	sudo cp mosquitto.conf /etc/mosquitto	Copiamo la configurazione di default

configurazione di base, possiamo aggiungere degli optional come per esempio una autenticazione per non permettere a chiunque di comunicare con il broker.

Per poter modificare questo aspetto chiudiamo il nostro broker digitando Ctrl+C nella shell e cambiamo la configurazione come segue:

```
persistence true
persistence_location /var/lib/mosquitto/
allow_anonymous false
password_file /etc/mosquitto/passfile
port 1883
listener 1884
protocol websockets
```

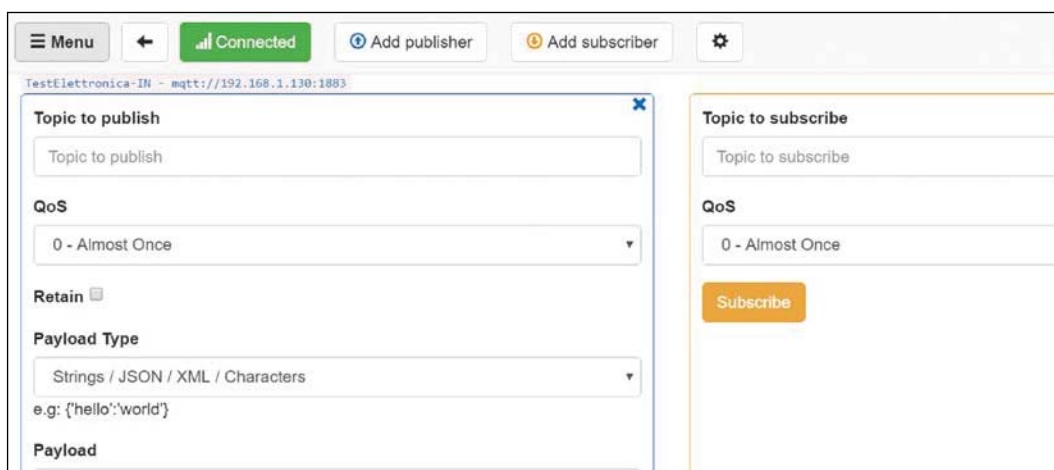
The screenshot shows the MQTTBox application window with the title bar 'MQTTBox' and menu options 'MQTTBox', 'Edit', and 'Help'. The main window has a 'Menu' button and a 'MQTT CLIENT SETTINGS' header. The settings are organized into two columns:

- MQTT Client Name:** Text input field containing 'TestElettronica-IN'.
- MQTT Client Id:** Text input field containing '2acc412f-a1c5-4591-80ff-06eb04918ee6' with a refresh icon on the right.
- Protocol:** Dropdown menu showing 'mqtt / tcp'.
- Host:** Text input field containing '192.168.1.130:1883'.
- Username:** Text input field containing 'Username'.
- Password:** Text input field containing 'Password'.
- Reconnect Period (milliseconds):** Text input field containing '1000'.
- Connect Timeout (milliseconds):** Text input field containing '30000'.
- Will - Topic:** Text input field containing 'Will - Topic'.
- Will - QoS:** Dropdown menu showing '0 - Almost Once'.

A blue 'Save' button is located at the bottom center of the settings area.

Fig. 7
MQTTBox: configurazione della connessione del client.

Fig. 8
MQTTBox in fase di
connessione attiva.



Dopo aver salvato e chiuso l'editor, digitiamo il seguente comando per aggiungere il file degli utenti e creare quindi il nostro primo utente:

```
sudo mosquitto_passwd -b /etc/mosquitto/passfile
elettronica_usr elettronica_pass
```

Ora abbiamo creato il nostro primo utente e possiamo riavviare il servizio:

```
sudo mosquitto -c /etc/mosquitto/mosquitto.conf
```

SERVER

Come nel capitolo precedente, anche a livello di server, o spazio hosting, per poter programmare in html e php ci sono diverse opzioni.

Se sei un neofita di questo mondo allora sicuramente i vantaggi di un hosting già pronto sono

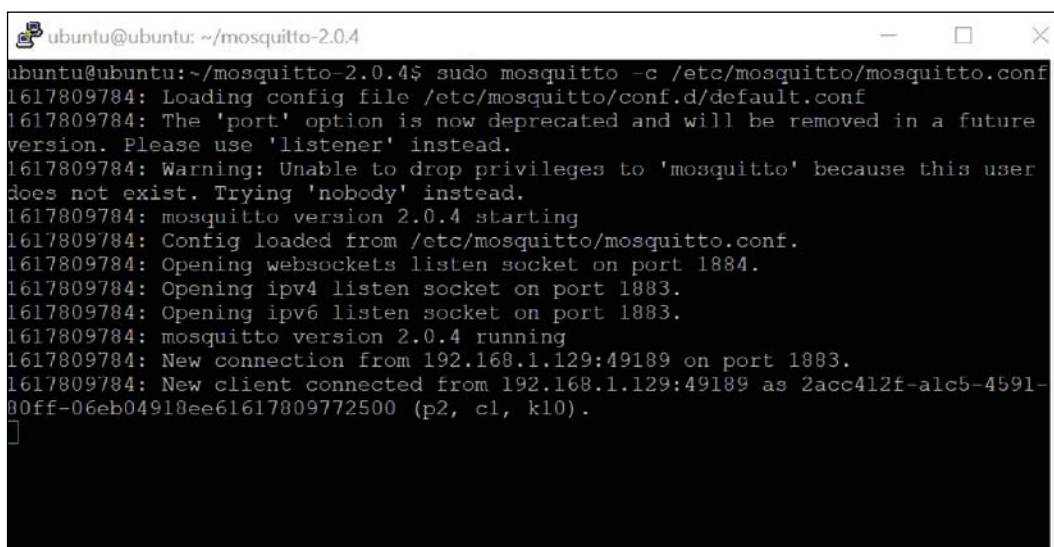
diversi. Oltre alle blasonate Aruba, Register o Host, ci sono numerosi portali che offrono piani gratuiti per potersi cimentare senza alcun costo iniziale, tra questi: Netsons o Altervista. Proprio quest'ultimo che abbiamo già citato ed esposto in passato, potrebbe fare al nostro caso.

In questo caso il broker, se non viene usato un broker pubblico al passo precedente, deve avere un indirizzo statico e bisogna intervenire anche nella configurazione del router per poter permettere l'uso di alcune porte.

Oltre a questa opzione, che non ha limiti dal punto di vista di scritture di pagine HTML e php, c'è anche la possibilità di aver il proprio webserver domestico, magari proprio a fianco del broker.

Sulla stessa Raspberry Pi, infatti, possiamo installare apache, uno dei webserver più utilizzati in ambiente Unix e non. In questa configurazione è

Fig. 9
Finestra SSH del
broker/Raspberry
Pi con lo stato della
connessione.



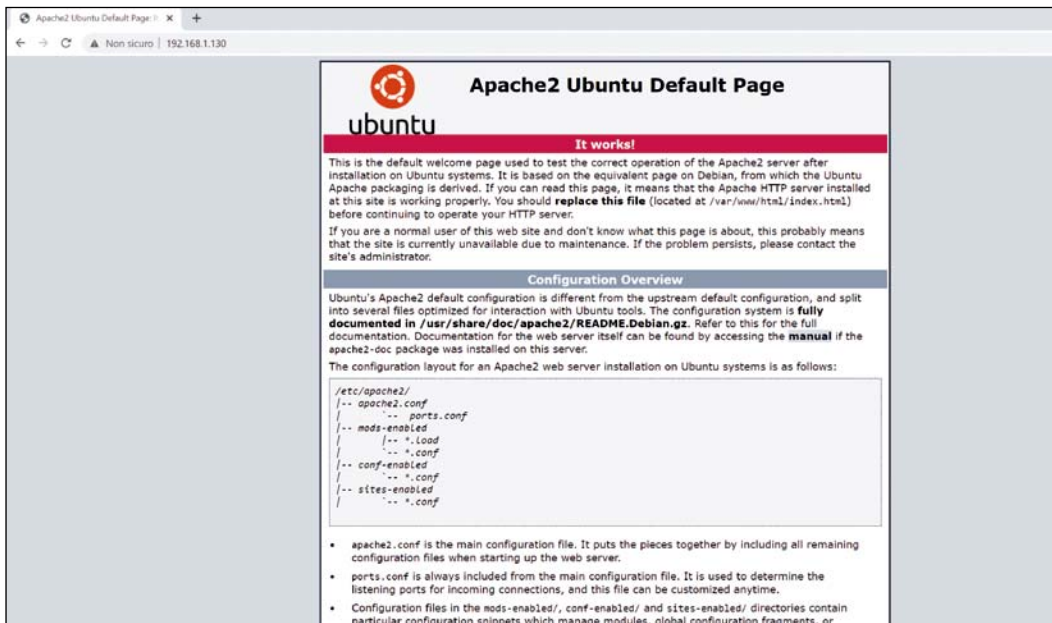


Fig. 10
Verifica
webserver
Apache su
Raspberry Pi.

possibile usare il broker locale e fare tutto all'interno della rete, oppure usare un broker pubblico, soluzione molto più scalabile e facilmente utilizzabile. Per fare ciò apriamo nuovamente una shell remota sul dispositivo attraverso Putty o qualunque programma usato in precedenza e digitiamo:

```
sudo apt update
sudo apt install apache2 -y
```

Per verificare se il pacchetto è stato correttamente installato, rechiamoci con un browser di un dispositivo all'interno della stessa rete e digiamo come URL l'indirizzo della Raspberry Pi, che nel caso della nostra applicazione d'esempio è 192.168.1.130, come si vede nella **Fig. 10**.

IL SOFTWARE

Creiamo all'interno della nostra root del server, indipendentemente che questo sia locale o remoto, una cartella che chiamiamo printer. All'interno di questa andiamo ad inserire i vari file del progetto. Prima fra tutti la nostra pagina, index.html. Questa è basata su due librerie molto importanti:

- JQuery (<https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js>);
- PahoMQTT (<https://cdn.jsdelivr.net/npm/paho-mqtt@1.0.1/mqttws31.js>).

Oltre a queste è presente un wrapper che contiene tutte le primitive per poter interagire con MQTT. Contiene infatti una serie di variabili e funzioni im-

portantissime che riducono di gran lunga il codice all'interno della pagina principale. All'interno di questa, come si può notare nel listato, questa libreria può essere divisa in tre parti: una prima nella quale vengono dichiarate le variabili globali, una che contiene la connessione al broker ed infine una per inviare e ricevere i messaggi (**Listato 1**).

Nella nostra main ci limitiamo a chiamare queste funzioni ed infatti la nostra pagina principale a livello di logica sarà simile a questo:

```
function onMessageArrive(data){
    console.log(''+data);
}
if(!MQTT_DANI.isConnected()){
    MQTT_DANI.connectBaseMQTT
    ('elettronica_usr', 'elettr_pass');
}
MQTT_DANI.setCbUpdateMessageArrive(onMessageArrive);
```

Per connetterci al broker e qualora volessimo comunicare ed inviare messaggi, richiamiamo l'istruzione:

```
MQTT_DANI.sendMsgUpdate({str:'stringa da stampare'});
```

FIRMWARE

Prima di procedere alla programmazione dovremmo impostare ArduinoIDE, ovvero il nostro ambiente di sviluppo, per poter scrivere correttamente sul modulo.



↓ Listato 1 - Corpo e funzioni della wrapper MQTT

```
var client = null;
var ip = "192.168.1.130";
var port = "8083";
var usessl = false;
var id_mqtt = (((1 + Math.random()) * 0x10000) | 0).toString(16).substring(1);
var username;
var password;
var topicUpdate = 'topicUpdate';
var topicCmd='topicCmd';
var callBackUpd = null;
var alreadyConnect = false;
var options = {
  userName: username,
  password: password,
  useSSL: usessl,
  onSuccess: onConnect,
  onFailure: onFailure
};
Una che contiene la connessione al broker:
function connectMQTT(user, pass, topic) {
  topicUpdate = topic;
  topicCmd = topic;
  options = {
    userName: user,
    password: pass,
    useSSL: usessl,
    onSuccess: onConnect,
    onFailure: onFailure
  };
  alreadyConnect = false;
  client = new Paho.MQTT.Client(ip, Number(port), id_mqtt);
  client.onConnectionLost = onConnectionLost;
  client.onMessageArrived = onMessageArrived;
  client.connect(getOptions());
}

function onConnect() {
  alreadyConnect = true;
  client.subscribe(topicUpdate, {qos: 0});
  if(cbConnected!=null){
    cbConnected();
  }
}
Ed infine una parte per inviare e ricevere i messaggi:
function onMessageArrived(message) {
  var value = message.payloadString;
  var datastream = message.destinationName;
  cbUpdateMessageArrive(value);
}

var cbUpdateMessageArrive = null;

function setCbUpdateMessageArrive(cb){
  cbUpdateMessageArrive = cb;
}

function sendMsgUpdate(data){
  try{
    data = JSON.stringify(data);
    var message = new Paho.MQTT.Message(data);
    message.destinationName = topicCmd;
    if(client.isConnected()){
      client.send(message);
    }else{
      connectMQTT();
      setCbConnected(function(){
        client.send(message);
        cbConnected = null;
      });
    }
  }catch(err){
    console.log(err);
  }
}
```

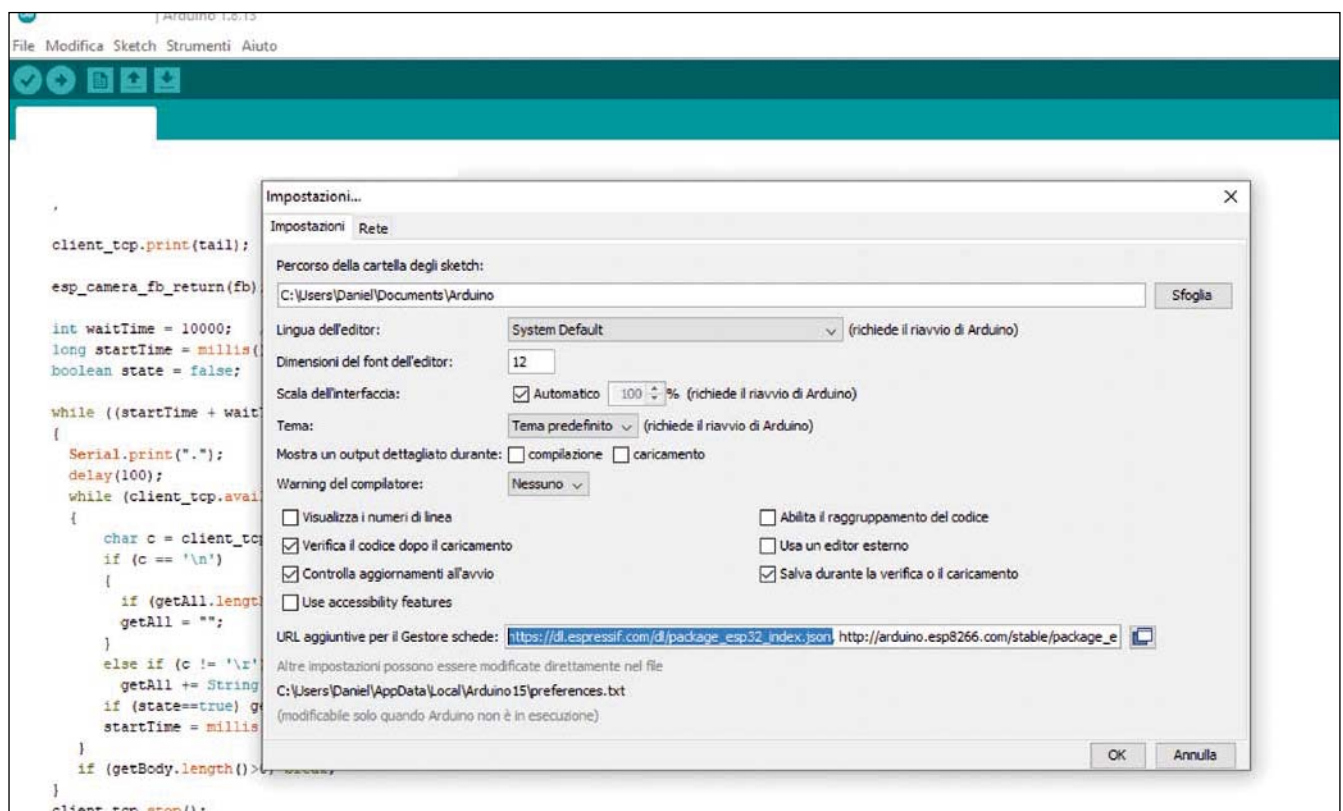


Apriamo quindi ArduinoIDE e andiamo in File->Impostazioni e aggiungiamo negli URL aggiuntivi per i gestori delle schede questo link http://arduino.esp8266.com/stable/package_esp8266com_index.json e clicchiamo sul pulsante OK (**Fig. 11**). Se si dispone già di uno o più URL si usa la virgola per concatenarli, proprio come nell'esempio. Ora scarichiamo correttamente il gestore della scheda: rechiamoci in Strumenti->Scheda: XXXX->Gestore schede..., cerchiamo "8266" e installiamo l'unico risultato della ricerca (**Fig. 12**). Uscendo selezioniamo nuovamente Strumenti->Scheda: XXXX e noteremo una nuova voce ESP8266 Modules, la quale al suo interno ha diverse altre scelte. Selezioniamo NodeMCU 1.0 (ESP-12E Module). Inoltre dobbiamo installare due librerie fondamentali per ottimizzare la comunicazione seriale: Adafruit Thermal printer library e SoftwareSerial. Per la prima è sufficiente recarsi in Sketch, all'interno di Arduino IDE, selezionare #include libreria dal menu a tendina che viene mostrato ed infine Gestione librerie. Si aprirà una finestra nella quale, come successo in precedenza è possibile una ricerca, nella quale

inseriremo Adafruit thermal printer. Ed infine clicchiamo su Installa. La seconda invece, SoftwareSerial, verrà installata da uno zip esterno. Scarichiamo lo zip da <https://github.com/plierup/espsoftwareserial>. Andiamo, come prima, alla voce Sketch di Arduino IDE e selezioniamo nuovamente #include libreria. Questa volta però clicchiamo sulla voce Aggiungi libreria da file .ZIP. Comparirà una schermata dalla quale selezioneremo il nostro ZIP appena scaricato. Ora abbiamo tutto il necessario per poter caricare il firmware sul dispositivo; colleghiamo quindi l'USB e selezioniamo la porta corretta e clicchiamo su Carica (**Fig. 13**). Vediamo ora come si compone il codice da uploadare sul nostro device suddiviso per le macro aree GLOBALI, FUNZIONE DI SETUP E FUNZIONE DI LOOP. Nella prima parte andremo a dichiarare tutti quelle variabili globali che useremo all'interno del programma:

```
#include "Adafruit_Thermal.h"
```

Fig. 11
Impostazione link nell'IDE Arduino.



📌 Listato 2 - Corpo centrale

```
//-----START SERIAL SECTION-----
SoftwareSerial mySerial(RX_PIN, TX_PIN);
Adafruit_Thermal printer(&mySerial);
AsciiWrapper nCode;

void castString(String cstr){
  int sz = cstr.length();
  int myInts[sz];
  printer.write(nCode.getCR());
  printer.write(nCode.getLF());
  for(int i=0;i<sz;i++){
    printer.write(nCode.cast(String(cstr[i], DEC).toInt()));
    if(DEBUG==1){
      Serial.println(cstr[i]);
      //Serial.println(cast(String(cstr[i], DEC).toInt()));
    }
  }
}
//-----END SERIAL SECTION-----

//-----START NETWORK SECTION-----
WiFiClient espClient;
PubSubClient client(espClient);
bool connectedNet = false;

void MQTTcallback(char* topic, byte* payload, unsigned int length) {
  if(DEBUG==1){
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
  }
  char myArr[length];
  for (int i = 0; i < length; i++) {
    myArr[i] = (char)payload[i];
  }
  String myString = String(myArr);
  DynamicJsonDocument doc(1024);
  DeserializationError error = deserializeJson(doc, myString);

  if(error){
    if(DEBUG==1){
      Serial.print("deserializeJson() failed: ");
      Serial.println(error.c_str());
    }
    return;
  }
  String fn = doc["str"];
  castString(fn);
}

void MQTTSubscribe(){
  String cmd = MYTOPIC;
  client.subscribe(cmd.c_str());
}

void MQTTreconnect() {

  int maxTry = 0;
  while (!client.connected()) {
    if(DEBUG==1){
      Serial.print("Attempting MQTT connection...");
    }
    String clientId = "MyRemotePrinter";
    //if (client.connect(clientId.c_str(), MQUSER.c_str(), MQPASS.c_str())) {
    if (client.connect(clientId.c_str())) {
      if(DEBUG==1){
        Serial.println("connected");
      }
      MQTTSubscribe();
    } else {
      if(DEBUG==1){
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
      }
    }
  }
}
```



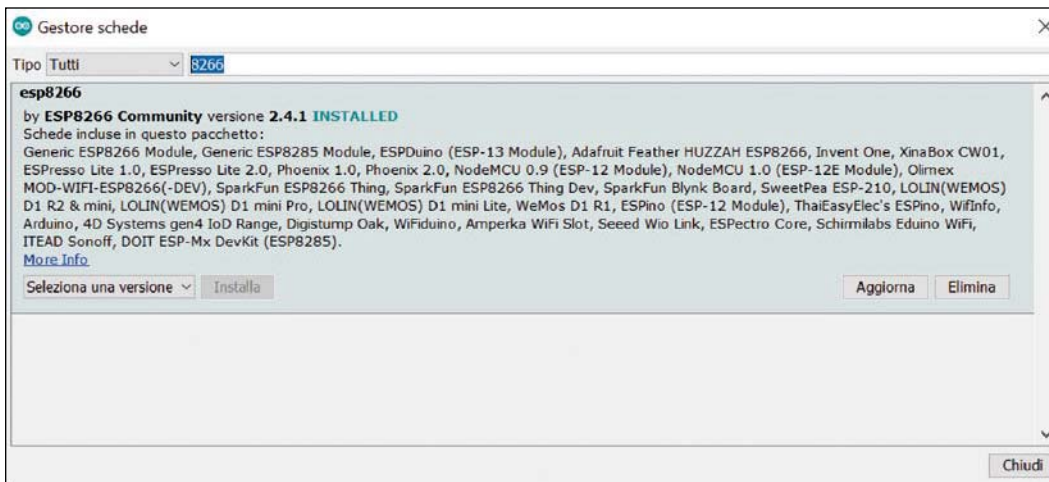


Fig. 12
Selezione della ESP8266 in Gestore Schede dell'IDE Arduino.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
#include <StreamUtils.h>
#include <PubSubClient.h>
#include "SoftwareSerial.h"
#include "AsciiWrapper.h"

//-----START DEFINE SECTION-----

#define TX_PIN D6
#define RX_PIN D5
#define BAUD 57600
String MYSSID="NET-SSID";
String MYPASS="NET-PASS";
#define MYTOPIC "testPrinter"
#define MYTOPICRES "testPrinterRes"
#define MQTTSERVER "192.168.1.130"
#define MQTTPORT 1883
#define MQTTBUFFERSIZE 2048
//String MQUSER="xxxxx";
//String MQPASS="xxxxx";
#define DEBUG 1
//-----END DEFINE SECTION-----
```

Come si può notare, dalla prima riga è stata aggiunta una libreria in C++: `AsciiWrapper.h`. Lo scopo di questa è rimappare i byte per poter dialogare con un dispositivo a 12V. Infatti potrebbero esserci interferenze nel caso andassimo a controllare un device con un tensione diversa da quella del nodemcu. Al posto di questa libreria è possibile inserire anche un dispositivo hardware nei PIN di TX e RX: il MAX3232. La seconda area si può suddividere a sua volta in

varie sottoaree come possiamo notare dal **Listato 2**. Qui la funzione di setup si apre con l'inizializzazione delle Seriali e successivamente cerca di connettersi ad una rete WiFi. Dopo questa fase si passa a creare il client per il protocollo MQTT: la callback che viene istanziata procede con la stampa di tutti i caratteri inviati. Infine vi è la macro parte di gestire il loop:

```
void loop() {
    if(connectedNet)
    {
        if (!client.connected())
        {
            MQTTreconnect();
        }
        client.loop();
    }
}

//-----END SETUP/LOOP SECTION-----
```

Questo è responsabile solo di tenere in piedi il canale inizializzato nella fase precedente.

VERIFICA

Per avviare il programma e controllare la buona

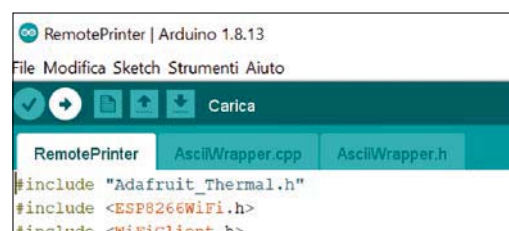


Fig. 13
Caricamento della libreria dalla finestra dell'IDE.

Listato 2 - Corpo centrale - Segue

```
        maxTry=maxTry+1;
        if(maxTry>=5){
            break;
        }
        delay(5000);
    }
}

bool connectAP(){
    WiFi.mode(WIFI_STA);
    WiFi.begin(MYSSID.c_str(), MYPASS.c_str());

    int ccount = 0;
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);

        if(DEBUG==1){
            Serial.print(".");
        }
        if(ccount==20){
            break;
        }
        ccount=ccount+1;
    }

    if(WiFi.status() == WL_CONNECTED){
        return true;
    }else{
        return false;
    }
}

bool connectWPS(){
    WiFi.mode(WIFI_STA);
    bool wpsSuccess = WiFi.beginWPSConfig();
    if(wpsSuccess) {
        String newSSID = WiFi.SSID();

        if(newSSID.length() > 0) {
            delay(500);
            return true;
        }else{
            delay(500);
            return false;
        }
    }
    delay(500);
    return false;
}
//----END NETWORK SECTION-----

//----START SETUP/LOOP SECTION-----
void setup() {

    if(DEBUG==1){
        Serial.begin(BAUD);
    }

    mySerial.begin(BAUD);
    printer.begin();
    connectedNet = connectAP();

    if(!connectedNet){
        connectedNet = connectWPS();
    }

    if(connectedNet){
        client.setServer(MQTTSERVER, MQTTPORT);
        client.setBufferSize(MQTTBUFFERSIZE);
        client.setCallback(MQTTcallback);
    }

    nCode = AsciiWrapper();
}
```

riuscita del nostro codice eseguiamo i seguenti passi:

- Verifichiamo in primis di aver acceso il servizio broker MQTT sul nostro Raspberry Pi:

```
sudo mosquitto -c /etc/mosquitto/mosquitto.conf
```

- Verifichiamo che entrambi i dispositivi siano sulla stessa rete.
- Digitiamo nella URL del browser di un terzo dispositivo connesso alla stessa rete il seguente indirizzo:

```
INDIRIZZO_RASPBERRYPI/printer/index.html?table=1
```

Ricordiamo il collegamento del nostro NodeMCU alla seriale, ossia:

- pin 6 (D6) del NodeMCU al 3 del connettore DB-9 della porta seriale;
- GND del NodeMCU al pin 5 del connettore DB-9 della porta seriale.

Bene, con questo riteniamo conclusa la descrizione della nostra applicazione.

CONCLUSIONI

Per spiegarvi come si può affacciare una periferica locale, come una stampante seriale, abbiamo preso spunto dall'esempio appena proposto della stampa diretta dell'ordinazione al ristorante da parte di un ipotetico cliente dotato di smartphone (trovate tutti i sorgenti per una piccola demo all'interno del nostro sito, www.elettronica.in.it). Partendo da questo è possibile sviluppare applicazioni che sfruttino il protocollo MQTT per accedere a risorse remote attraverso Internet e una connessione WiFi. □



Cosa occorre?

La board con ESP8266 utilizzata in questo progetto (cod. NODEMCUESP) è in vendita a Euro 9,50. 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>

Strumenti di misura per veri professionisti!



Rilevatore da parete 3 in 1 (legno, metalli, cavi elettrici)

Rileva la presenza di cavi elettrici sotto tensione (con passaggio di corrente), strutture in legno (telai, travetti, barre, ecc), oggetti metallici (travi, telai, tubi, tondini, perni, barre in metallo). Dispone di ampio display retroilluminato, buzzer di segnalazione, pulsante di calibrazione automatica. Si spegne automaticamente dopo 3 minuti di inutilizzo. Alimentazione con batteria da 9 V (acquistabile separatamente), dimensioni 156x70x26 mm.

€29,50

Cod. FR760

Rilevatore da parete professionale (legno, metalli, cavi elettrici)

Rileva la presenza di cavi elettrici sotto tensione (con passaggio di corrente), strutture in legno (telai, travetti, barre, ecc), oggetti metallici (travi, telai, tubi, tondini, perni, barre in metallo). Dispone di ampio display LCD retroilluminato, buzzer di segnalazione, spegnimento automatico dopo 5 minuti di inutilizzo. Alimentazione tramite batteria al litio (integrata) da 3,7 volt 1500 mAh. Dimensioni 195x90x35 mm.



- ✓ Professionale
- ✓ Batteria al litio ricaricabile integrata da 3,7 V - 1500 mAh con autonomia fino a 10 ore
- ✓ Case robusto
- ✓ Prestazioni migliorate

€78,00

Cod. FR801

Misuratore di distanza fino a 80 m con LASER verde e messaggi vocali.

Metro LASER multifunzione compatto con display LCD retroilluminato, messaggi vocali, menu a icone, livella digitale, memoria interna (in grado di memorizzare fino a 100 misurazioni) e batteria ricaricabile agli ioni di litio. È in grado di effettuare le seguenti misurazioni: distanza (da 20 cm a 80 metri), area, volume, misurazione pitagorica, misurazione verticale e molte altre. Visualizza lo stato di carica delle batterie; si spegne automaticamente se inattivo (laser dopo 30 secondi, strumento dopo 2 minuti). Grado di protezione IP54. Dimensioni 120x50x26 mm.



€79,00

Cod. FR798

Misuratore di distanza LASER da 20 cm a 30 m

Compatto metro LASER con display LCD retroilluminato. Permette di misurare in modo estremamente facile e veloce la distanza (da 20 cm a 30 metri), l'area e il volume. Dispone di spegnimento automatico del LASER e del dispositivo; visualizza anche lo stato di carica delle batterie. Alimentazione con 2 batterie mini stilo da 1,5 volt tipo AAA (incluse). Dimensioni 105x34x20 mm



€27,00

Cod. FR797

Prezzi IVA inclusa

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

€ 11,50

Cod. **BOXVGATOHDMI**

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.

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**

MINIUPS: CONTINUITÀ PER IL LAMPONE



degli Ingg. **ANDREA RIBA**
e **FABIO FILIPPA**

Piccolo gruppo di continuità per Raspberry Pi in grado di mantenere accesa la board in caso di blackout .
Seconda ed ultima puntata.

Ci siamo lasciati, nel fascicolo precedente, dopo avervi introdotto e descritto una scheda contenente un piccolo gruppo di continuità per la Raspberry Pi, molto interessante non solo perché consente di mantenere accesa la board anche in mancanza della tensione di rete (superando eventuali blackout) per un certo tempo, ma anche perché nel funzionamento in blackout, al rilevamento dello stato di batteria quasi scarica, provvede a dialogare con la Raspberry Pi per fare in modo che questa arresti i processi in corso in maniera sicura e senza perdere dati o danneggiare la SD-Card. Ora, dopo aver descritto la configurazione e le prime operazioni di interfacciamento con il bus I²C in locale da shell bash, riguardanti l'accesso ai registri e all'area utente della memoria EEPROM, passiamo a un utilizzo più avanzato che comprende la realizzazione di script Python per la gestione di funzionalità.

Listato 1

```
#!/usr/bin/python
# -----
# Prerequisiti
# sudo apt-get install python-smbus python-dev
#
# Determinare numero bus I2C
# ls -l /dev/i2c-1
# crw-rw---- 1 root i2c 89, 1 Feb 14 2019 /dev/i2c-1
# -----
import os
import sys
import smbus # Per gestione bus I2C
import time
UP_DEVICE_ADDRESS = 0x08 # Indirizzo microcontrollore
UP_REG_ADDRESS = 0x00 # Registro microcontrollore
# -----
# MAIN
# -----
if __name__ == "__main__":
    # Pulisce lo schermo
    os.system("clear")
    # BUS I2C-1 (vedere risposta al comando ls -l /dev/i2c-1)
    rpi_bus = smbus.SMBus(1)
    cycle_counter = 0
    while True:
        try:
            cycle_counter+=1;
            print("-----")
            sys.stdout.flush()
            print("Ciclo lettura-incremento-scrittura-lettura numero: " + \
                  str(cycle_counter))
            sys.stdout.flush()
            print("Premere CTRL+C per uscire.")
            sys.stdout.flush()
            print("-----")
            sys.stdout.flush()
            # Esegue la lettura del registro specificato
            byte_val = rpi_bus.read_byte_data(UP_DEVICE_ADDRESS, UP_REG_ADDRESS)
            print( "Lettura registro : " + str(UP_REG_ADDRESS) + \
                  " - Valore letto: " + str(byte_val))
            sys.stdout.flush()
            time.sleep(1)
            # Incremento di 1 il valore del dato letto
            byte_val += 1
            print("Incremento il valore letto di 1")
            sys.stdout.flush()
            # Esegue una scrittura singola sul registro specificato
            print( "Scrittura Registro: " + str(UP_REG_ADDRESS) + \
                  " - Valore scritto: " + str(byte_val))
            sys.stdout.flush()
            rpi_bus.write_byte_data(UP_DEVICE_ADDRESS, UP_REG_ADDRESS, byte_val)
            time.sleep(1)
            # Esegue una nuova lettura dello stesso registro
            byte_val = rpi_bus.read_byte_data(UP_DEVICE_ADDRESS, UP_REG_ADDRESS)
            print( "Lettura registro : " + str(UP_REG_ADDRESS) + \
                  " - Valore letto: " + str(byte_val))
            sys.stdout.flush()
            # Pausa prima del prossimo ciclo
            time.sleep(5)
        except (KeyboardInterrupt, SystemExit), ex:
            # Gestisce interruzione CTRL+C
            print("CTRL-C detected!")
            sys.stdout.flush()
            sys.exit(0)
        except Exception as ex:
            # Gestione qualsiasi altra eccezione
            template = "ERROR: An exception of type {0} occurred." + \
                       "Arguments:\n{1!r}"
            msg = template.format(type(ex).__name__, ex.args)
            print(msg)
            sys.stdout.flush()
            # Codice di uscita
            print("(main) Exiting from main program")
            sys.stdout.flush()
            sys.exit(0)
```


GESTIONE I²C DA SCRIPT PYTHON SU RASPBERRY PI

Dunque, verificato il funzionamento del sistema attraverso il terminale della Raspberry Pi ci addentriamo più nella programmazione lato utente, per vedere come poter implementare la lettura e la scrittura dei registri del microcontrollore direttamente da Python. Prima di procedere è opportuno installare alcuni tool utilizzando il seguente comando fornito con privilegi di amministratore:

```
sudo apt-get install -y python-smbus python3-smbus
python-dev python3-dev
```

l'opzione `-y` evita all'utente di dover rispondere Y ad ogni pacchetto che deve essere installato. Questi pacchetti permettono la gestione del bus I²C della Raspberry Pi da script scritti per essere eseguiti con interprete Python 2.x e Python 3.x. La libreria utilizzata per la gestione del bus I²C da script Python è la *smbus*, di seguito si riporta un esempio di script che illustra come utilizzare i metodi di lettura e scrittura sui registri del microcontrollore (che, si ricorda, dal punto di vista della Raspberry Pi si comporta come *slave device*), il codice presentato non fa altro che incrementare di una unità, ogni 5 secondi, il valore contenuto in un registro del microcontrollore; per fermare l'esecuzione dello script è sufficiente premere la combinazione di tasti CTRL+C.

Per eseguire lo script si scrive nella *shell* il seguente comando seguito dalla pressione del tasto *Invio*.

```
python esempio-i2c.py
```

Il relativo codice script `esempio-i2c.py` è riportato nel **Listato 1**.

Per trasferire il file da un sistema Windows sulla Raspberry Pi attraverso una connessione di rete si può utilizzare ad esempio l'applicazione WinSCP, che permette il trasferimento di file con protocollo SFTP (SSH File Transfer Protocol) attraverso un'interfaccia grafica anche con supporto *drag&drop*, inoltre, una volta che i file sono stati trasferiti, ne permette anche la modifica con un semplice clic. Come editor Windows si suggerisce Notepad++ in quanto salva i file in formato direttamente compatibile con sistemi operativi Linux come quello utilizzato sulla Raspberry Pi.

L'esempio riportato comprende anche altri aspetti generali sulla programmazione applicati più nello specifico al linguaggio Python come la gestione delle eccezioni, questo proprio solo per rendere

più completo il codice ed eventualmente fornire qualche spunto in tal senso, di seguito invece ci si concentrerà prevalentemente su quelli che sono gli aspetti più di nostro interesse ovvero la comunicazione sul bus I²C.

Dal punto di vista della programmazione Python la prima cosa da fare consiste nel comunicare all'interprete Python che nel proseguo dello script dovranno essere utilizzati metodi specifici (relativi al bus I²C) contenuti in una certa libreria (*smbus*), a tale scopo si ha la seguente riga di codice:

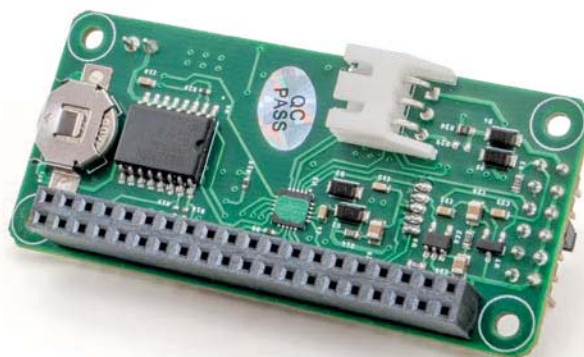
```
import smbus
```

Per utilizzare questi metodi (che sono in sostanza delle funzioni) si deve istanziare un oggetto di tipo bus il cui codice si trova all'interno della libreria SM-Bus.

Un'istanza, da un punto di vista pratico, è un oggetto fisico (quindi rappresentato dal codice che viene effettivamente caricato in memoria ed



La scheda MiniUPS: si applica alla Raspberry Pi mediante un connettore tipo header.



inizializzato) ricavato da un modello che è contenuto in una specifica libreria (in questo caso *smbus*) e che viene inizializzato proprio per essere utilizzato in seguito, in questo caso l'istanza del bus sarà creata con la seguente riga di codice:

```
rpi_bus = smbus.SMBus(1)
```

da notare come il nome assegnato *rpi_bus* sia completamente arbitrario, può quindi essere un qualsiasi nome assegnato in modo significativo atto a ricordare ciò che rappresenta realmente in quanto punterà e permetterà l'accesso all'oggetto vero e proprio presente nella memoria del sistema fino a quando il codice dello script sarà in esecuzione; ad ogni modo, qualunque esso sia, sarà comunque il riferimento con cui andremo ad identificare, all'interno del nostro script, il bus I²C fisico della Raspberry Pi.

Come illustrato in precedenza, nel caso della gestione manuale del bus I²C attraverso la shell bash della Raspberry Pi, si deve indicare su quale bus I²C si vuole lavorare, questo perché, in certi sistemi, è possibile che siano presenti più bus I²C, nel caso in esame il bus è unico ed è indirizzato con 1. Per determinare il riferimento del bus si può utilizzare il seguente comando da shell:

```
ls -l /dev/i2c-*
```

In questo caso, trovando solo un unico riferimento, *i2c-1*, si utilizza appunto 1 come identificatore del bus.

Sul bus I²C si possono avere più dispositivi (detti *slave*) che potranno essere indirizzati dalla Raspberry Pi (che in questo caso ha il ruolo di *master* ovvero di coordinatore delle operazioni che saranno effettuate sul bus I²C), è quindi utile definire delle costanti relative all'indirizzo di ogni slave che, nel nostro caso, è costituito dal microcontrollore presente sulla scheda con indirizzo 0x08, quindi:

```
UP_DEVICE_ADDRESS = 0x08
```

Per eseguire una lettura su un singolo registro si utilizza il metodo *read_byte_data* sull'istanza del bus che è stata identificata con il nome *rpi_bus*, in particolare:

```
byte_val = rpi_bus.read_byte_data(UP_DEVICE_ADDRESS,  
UP_REG_ADDRESS)
```

in cui:

- **UP_DEVICE_ADDRESS** : indirizzo dello *slave device* (in questo caso il microcontrollore);
- **UP_REG_ADDRESS** : indirizzo del registro che si desidera leggere (nell'esempio 0x00 in EEPROM);
- **byte_val**: variabile che conterrà il valore restituito dalla lettura del registro **UP_REG_ADDRESS**.

In modo del tutto analogo, per scrivere su un registro dello *slave device*, occorrerà fornire l'indirizzo dello slave sul bus I²C (quindi il microcontrollore), l'indirizzo del registro in cui si vuole scrivere (nell'esempio si scrive sull'area EEPROM all'indirizzo 0x00) e il dato da scrivere.

La sintassi da utilizzare è la seguente:

```
rpi_bus.write_byte_data(UP_DEVICE_ADDRESS, UP_REG_  
ADDRESS, byte_val)
```

in cui:

- **UP_DEVICE_ADDRESS** è l'indirizzo dello *slave device* (in questo caso il microcontrollore);
- **UP_REG_ADDRESS** è l'indirizzo del registro dello *slave device* in cui si vuole scrivere (nell'esempio 0x00 in EEPROM);
- **byte_val** è la variabile che conterrà il valore da scrivere nel registro **UP_REG_ADDRESS**.

Questo, in sintesi, è quanto serve per poter gestire direttamente da uno script Python la lettura e la scrittura su bus I²C, ulteriori informazioni sono reperibili consultando l'help del pacchetto *smbus* e la relativa documentazione Python.

GESTIONE LINEA RPIOFF DA SCRIPT PYTHON SU RASPBERRY PI

La linea RPIOFF è gestita dalla scheda per segnalare quando è previsto uno spegnimento imminente del sistema a causa di un black-out e della condizione di batteria quasi scarica. Si tratta di un segnale digitale, facilmente interpretabile da Raspberry Pi. Questa linea è collegata direttamente al pin GPIO17 del connettore GPIO ed è l'unico pin dedicato utilizzato dalla scheda.

Il tempo previsto tra la segnalazione dell'evento di spegnimento (linea RPIOFF) e l'effettivo spegnimento della Raspberry Pi è impostato a 60 secondi. L'autonomia della batteria tampone, quando questa risulta completamente carica ed in completa assenza della tensione di alimentazione primaria, risulta maggiore di 4 minuti considerando anche una condizione operativa gravosa da parte della



Listato 2

```
#!/usr/bin/python
# -----
# Servizio gestione shutdown da linea RPIOFF
# -----
import RPi.GPIO as GPIO
import time
import os
import sys
from datetime import datetime
import subprocess

# Determina la data corrente
now = datetime.now()
date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
# Definisce file Log in modalita append, e flush immediato
LogFilename = "/home/pi/rpimups_shsrv.log"
logFile = open(LogFilename, 'a', 0)
logFile.write('rpimups_shsrv started on: ' + date_time + '\n')
# Disattiva warnings GPIO
GPIO.setwarnings(False)
# Imposta modalita' utilizzo GPIO
GPIO.setmode(GPIO.BCM)
# Definisce le linee GPIO
GPIO.setup(17, GPIO.IN) # GPIO17 input (RPIOFF)
flag = False
try:
    while (True):
        # GPIO17: gestione ingresso RPIOFF per shutdown
        if (GPIO.input(17) == 1):
            # Attende 5 secondi dopo che la linea
            # RPIOFF e' stata attivata
            counter = 5
            time.sleep(1)
            GPIO.cleanup()
            # Scrive sul Log
            now = datetime.now()
            date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
            logFile.write('rpimups_shsrv fire shutdown on: ' + date_time + '\n')
            logFile.close()
            time.sleep(0.5)
            # Comando di shutdown
            subprocess.call(["shutdown", "-h", "now"])
            break
except Exception as ex:
    print('Exception detect!')
    sys.stdout.flush()
    template = "An exception of type {0} occurred. Arguments:\n{1!r}"
    message = template.format(type(ex).__name__, ex.args)
    print(message)
    sys.stdout.flush()
finally:
    sys.exit(0)
```

Raspberry Pi che preveda un assorbimento in corrente dell'ordine dei 3 A. La batteria fornita a supporto con la scheda è comunque in grado di erogare una corrente di 15C, corrispondente a circa 6.300 mA, per un tempo minimo di circa 3,8 minuti (questo ipotizzando una batteria nuova, completamente carica) il che garantisce ai normali livelli di utilizzo una operatività più che sufficiente per uno spegnimento controllato nei tempi indicati.

Per gestire in sicurezza lo shutdown è fondamentale intercettare il momento in cui la linea RPIOFF viene forzata al livello alto dal modulo UPS, questo si può fare con due approcci differenti:

- controllare in polling (modo ciclico) dalla propria applicazione lo stato della linea RPIOFF e nel caso in cui si rilevi lo stato attivo provvedere allo shutdown del sistema;
- Installare un servizio che, indipendentemente dalle applicazioni che sono in esecuzione, lavo-

rando in background avvisi tutto il sistema che deve essere eseguito uno shutdown;

Entrambi i metodi hanno lo scopo di effettuare lo shutdown della Raspberry Pi, ma si hanno importanti differenze.

Nel primo è responsabilità dell'utente gestire in modo ciclico la linea RPIOFF, nel secondo è il sistema che si preoccupa di gestire la condizione di shutdown in modo indipendente dai programmi utente in esecuzione.

Il primo approccio non sarà descritto nel seguito di questo articolo, il codice di esempio è comunque disponibile sul repository GitHub (file `ex-polling-GPIO17.py`), si affronterà invece in modo più approfondito il secondo metodo che prevede l'utilizzo di un servizio appositamente concepito allo scopo.

APPROCCIO MEDIANTE SERVIZIO DEDICATO ESEGUITO IN BACKGROUND

Il controllo attraverso servizio dedicato, eseguito automaticamente alla partenza, come già anticipato, è il modo più sicuro per gestire lo spegnimento sicuro del sistema in quanto non è necessario aggiungere tale gestione ad un eventuale programma utente, ma il servizio, lavorando in background (quindi in totale trasparenza all'utente) provvede in autonomia a controllare ed a spegnere in modo sicuro il sistema in caso di necessità.

Per definire un servizio, nei sistemi operativi più recenti basati su *systemd*, sono necessari due file distinti.

Il primo tipo di file è utilizzato per descrivere la risorsa che in questo caso è un servizio (questi file in *systemd* sono chiamati *unit* ed hanno estensione *.service*). L'altro file è quello che implementa il codice vero e proprio del servizio eseguito dal sistema. Infine, per rendere il tutto operativo, si deve effettuare l'installazione del servizio che, da un punto di vista pratico, si traduce nell'eseguire

alcuni comandi nella shell per fare sì che questo possa essere avviato automaticamente al *boot* della Raspberry Pi.

Anche in questo caso il codice relativo al servizio sarà presentato come script Python, ma potrebbe essere scritto in un qualsiasi altro linguaggio di programmazione; il file *.service* utilizzato invece per descrivere il servizio sarà un semplice file di testo strutturato in modo particolare, suddiviso in sezioni ben definite, ciascuna con un suo particolare scopo e semantica.

Iniziamo con il presentare il codice dello script Python (`rpimups_shsrv.py`) che andrà ad implementare il controllo e la gestione vera e propria del segnale RPIOFF: è riportato nel **Listato 2**.

Il codice dello script che implementa il servizio è anch'esso piuttosto semplice; per tracciare l'avvio e l'esecuzione dello shutdown è prevista la scrittura di un file di log `rpimups_shsrv.log` nella cartella `/home/pi`, in seguito si definisce la linea GPIO17 come ingresso.

Dopodiché si ha un ciclo infinito che si occupa di controllare costantemente lo stato della linea GPIO17, se questo risulta alto, dopo un intervallo di tempo di 5 secondi lo script registra l'evento sul file di log e chiude il file, in seguito viene eseguito il comando di shutdown `-h now` utilizzando *subprocess*, a questo punto lo script termina.

Come anticipato, un servizio deve essere corredato di uno unit file che ne descriva alcune caratteristiche; nel **Listato 3** riportiamo il contenuto del file `rpimups_shsrv.service` per il servizio in esame. Nella sezione [Unit] il parametro Description contiene una semplice descrizione del servizio a cui l'unità si riferisce. Il parametro StartLimitIntervalSec fa sì che *systemd* possa fare ripartire sempre il servizio, più in dettaglio, come impostazione predefinita, *systemd* fa ripartire un servizio se quest'ultimo non ha già eseguito più di 5 tentativi in un intervallo di tempo di 10 secondi, se si eccede nel numero di tentativi il servizio non viene più fatto partire. Impostando `StartLimitIntervalSec=0` si evita questo comportamento imponendo a *systemd* di fare ripartire sempre il servizio.

La sezione [Service] con la voce `Restart=always` specifica che il servizio deve essere fatto sempre ripartire in presenza di una sua uscita (qualunque sia il motivo), con `RestartSec=1` che prima di riavviarlo deve attendere 1 secondo, questo per non sovraccaricare il sistema (per impostazione predefinita, il riavvio avviene dopo circa 100 ms); in seguito si trova la riga che contiene il riferimento al codice vero e proprio del servizio.

Listato 3

```
[Unit]
Description=Shutdown manager
StartLimitIntervalSec=0
[Service]
Restart=always
RestartSec=1
ExecStart=/usr/bin/python2 /home/pi/rpimups_shsrv.py
[Install]
WantedBy=multi-user.target
```



Avendo uno script python questo dovrà essere eseguito attraverso l'interprete Python ed il codice dello script sarà presente al percorso **/home/pi**. Questa sezione della unit non è interpretata da *systemd* in runtime, ma è utilizzata dai comandi di installazione (enable) e rimozione (disable) del servizio. L'ultima sezione [Install], con WantedBy specifica per quale tipologia di stato di attività della macchina il servizio deve essere eseguito. Completata l'analisi del codice che implementa le operazioni che saranno svolte dal servizio e della *unit* che lo descrive si deve procedere con l'installazione del servizio nel sistema. Di seguito la procedura con i comandi da utilizzare (tutti seguiti dalla pressione del tasto *Invio*). Ci si posiziona nella cartella **/home/pi**:

```
cd /home/pi
```

si scarica il repository con git:

```
sudo apt-get update
sudo apt-get -y install git
git clone http://github.com/Italsensor/RPI-MINI-UPS-R0
```

ci si sposta nel seguente percorso:

```
cd /home/pi/RPI-MINI-UPS-R0/Examples/Shutdown/Service
```

si impostano i permessi di esecuzione con il seguente comando:

```
chmod 700 rpimups_shsrv.py ; chmod 700 rpimups_shsrv.service
```

Si copia il file di unit del servizio nella cartella **/etc/systemd/system**, il comando deve essere eseguito con privilegi di *root*:

```
sudo cp rpimups_shsrv.service /etc/systemd/system/rpimups_shsrv.service
```

Si copia il codice che sarà eseguito dal servizio nella cartella **/home/pi**:

```
sudo cp rpimups_shsrv.py /home/pi
```

Prima di avviare il servizio consigliamo di controllare che il modulo UPS abbia l'alimentazione principale applicata, onde evitare un possibile shutdown non previsto; infatti tenete presente che una volta avviato, se per qualche motivo la linea RPIOFF

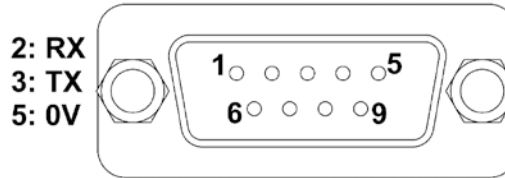


Fig. 1
Connettore maschio 9 poli porta RS232 lato PC (vista lato pin).

dovesse risultare attiva, trascorso l'intervallo di tempo predefinito sarebbe eseguito lo shutdown del sistema con conseguente spegnimento della Raspberry Pi.

Fatti i debiti controlli, si può quindi procedere con l'avviare il servizio, utilizzando allo scopo il seguente comando:

```
sudo systemctl start rpimups_shsrv.service
```

Per completezza di informazione, nel caso delle *unit* che descrivono dei servizi, specificare l'estensione **.service** non è strettamente necessario in quanto, per impostazione predefinita, *systemctl* considera che l'estensione sia proprio **.service**. Per controllare lo stato del servizio si utilizza il comando:

```
sudo systemctl status rpimups_shsrv
```

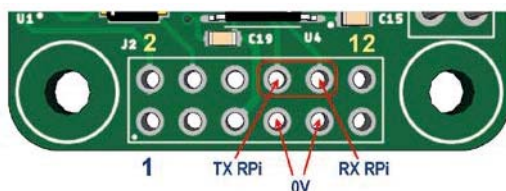
Si rende lo script avviabile al *boot* del sistema:

```
sudo systemctl enable rpimups_shsrv
```

questo comando provvede a creare un link simbolico dalla copia del file del servizio contenuto in **/lib/systemd/system** oppure in **/etc/systemd/system/** nella cartella che contiene i riferimenti dei file che *systemd* deve lanciare al boot del sistema (tipicamente in **/etc/systemd/system/nome_target.target.wants**).

Un appunto merita essere fatto circa questa differenziazione tra le cartelle che contengono i file **.service**: la cartella **/lib/systemd/system** (questo percorso è valido per distribuzioni Debian e derivate come nel caso del Raspberry, in altre distribuzioni derivate ad esempio da RedHat come CentOS il percorso è **/usr/lib/systemd/system**) è utilizzata dai file che sono forniti come pacchetti scaricabili dai vari repository, mentre la cartella **/etc/systemd/system/** contiene i file gestiti direttamente dall'amministratore di sistema (quindi in questo caso da noi) ed è per questo motivo che il file è stato copiato in quest'ultima specifica cartella.

Fig. 2
Segnali della porta
seriale RS232
sulle piazzole del
connettore J2.



In modo analogo, per rimuovere l'esecuzione del servizio al *boot* (quindi eliminare automaticamente il link simbolico creato in precedenza con il comando *enable*), si utilizza il comando:

```
sudo systemctl disable rpimups_shsrv
```

Per completezza riportiamo qui di seguito altri comandi utilizzabili per la gestione dei servizi. Il primo di essi è quello che permette di arrestare il servizio:

```
sudo systemctl stop rpimups_shsrv
```

Quello seguente, invece, è quello per fare ripartire il servizio arrestato:

```
sudo systemctl restart rpimups_shsrv
```

Per maggiori dettagli sul funzionamento di *systemd* e sull'utilizzo del relativo daemon di gestione *systemctl* si rimanda al manuale on-line.

Per un riassunto dei passi principali sopra riportati, relativi alla creazione di un servizio, può essere utile consultare la guida ufficiale presente sul sito dell'organizzazione Raspberry Pi (<https://www.raspberrypi.org>).

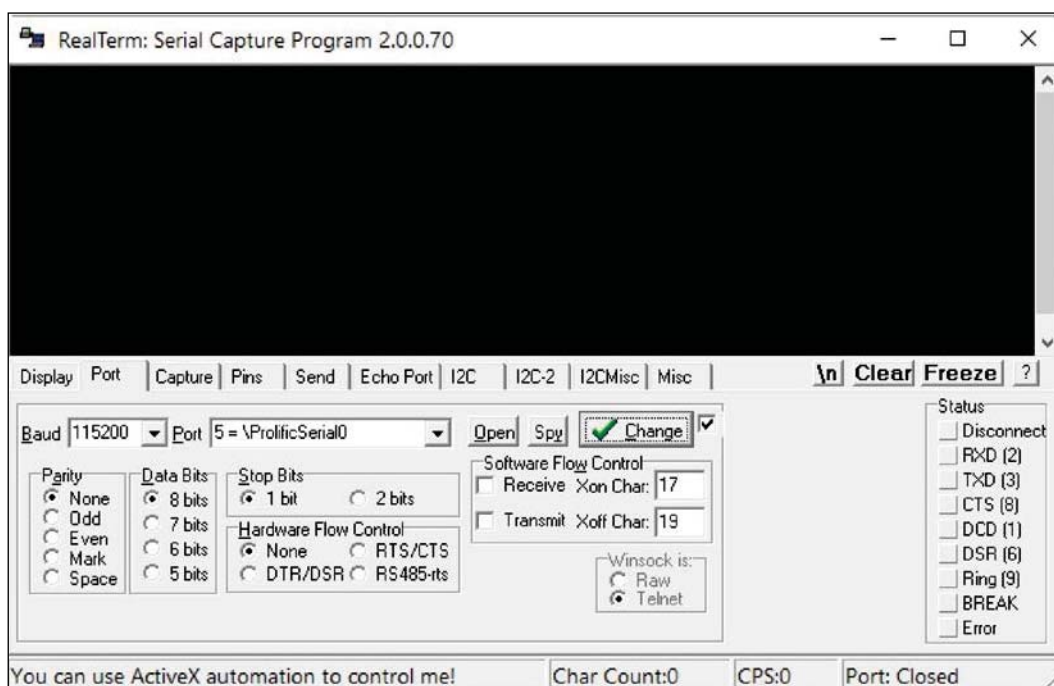
GESTIONE PORTA SERIALE SU RASPBERRY PI

La scheda è anche equipaggiata con driver seriale RS232 (MAX3232) per permettere un interfacciamento diretto tra le linee TX e RX della Raspberry Pi presenti sul connettore GPIO (rispettivamente pin 8 e pin 10) ed il mondo esterno. Il connettore a vaschetta 9 poli presente sui PC più datati oppure sugli adattatori USB/RS232 ha la piedinatura mostrata nella **Fig. 1**. Questa disposizione corrisponde anche al connettore femmina, visto però dal lato saldature del connettore stesso (lato dove si saldano i cavetti al connettore).

In riferimento alla connessione in esame, quello che ci interessa sono le sole linee RX e TX, oltre alla massa (GND) ovvero lo 0V, in quanto la connessione che si andrà a realizzare è la classica 115200,8,N,1 senza handshake nè gestione dei segnali di controllo della comunicazione (DTR, RTS, CTS ecc.) solo come esempio per illustrare come effettuare la connessione e la comunicazione sulla porta seriale RS232.

Facendo riferimento alla disposizione dei pad sul PCB per il connettore J2, riportato in **Fig. 2** per co-

Fig. 3
Configurazione
dati per la
connessione
seriale.



modità, la connessione fisica alla nostra Raspberry Pi sarà effettuata utilizzando il pin 8 (TX), il pin 10 (RX) e indifferentemente il pin 7 oppure 9 per la connessione dello OV.

Siccome il connettore dal lato Personal Computer è solitamente di tipo maschio (perché il PC è un DTE), come anche il connettore che si trova sui numerosi modelli di convertitore USB/RS232 in commercio, la connessione dovrà essere effettuata intestando lato Raspberry Pi un connettore a vaschetta corrispondente, ma di tipo femmina; quindi serve un connettore DB-9 femmina.

La connessione, affinché possa essere in grado di scambiare dati tra la Raspberry Pi e il sistema esterno (in questo caso si utilizza un PC con un adattatore USB) dovrà essere di tipo incrociato (cross) quindi i segnali RX e TX dal connettore J2 dovranno andare rispettivamente sui pin TX e RX del connettore femmina che andrà ad innestarsi sul maschio a bordo del PC.

Per la comunicazione su PC si può utilizzare un qualsiasi terminale seriale; in questo caso sarà utilizzato **RealTerm** (Fig. 3).

Una volta in esecuzione, si seleziona il pannello **Port** e si configurano i dati di connessione come riportato nella Fig. 3.

Per la selezione della porta di comunicazione (Port), avendo utilizzato un adattatore USB to RS232, si deve identificare quale è il numero di porta (VCP = Virtual Com Port) assegnato dal sistema, nel caso in esempio è la COM5.

Per identificare il numero di VCP su Windows si deve accedere al pannello Gestione dispositivi; un modo per fare ciò è premere la combinazione di tasti Windows+X selezionando la voce di menu **Gestione dispositivi**, ma nulla vieta di accedere dal menu principale (menu di avvio).

Una volta determinato il numero di porta assegnato dal sistema all'adattatore (vedete un esempio nella Fig. 4) lo si seleziona nel terminale di comunicazione seriale, a questo punto si può procedere con l'apertura della comunicazione, premendo il pulsante **Open**.

Per visualizzare i caratteri digitati nel terminale, che saranno quindi trasmessi alla Raspberry Pi, si seleziona il pannello **Display** configurando le impostazioni come riportato nella Fig. 5 (selezione opzione Half Duplex).

Completate le impostazioni lato PC ci si sposta sul Raspberry. Dal repository si può scaricare il file **ex-serial.py** (per Python 2.x) oppure **ex-serial-py3.py** (per Python 3.X), il funzionamento è identico per entrambe le versioni.

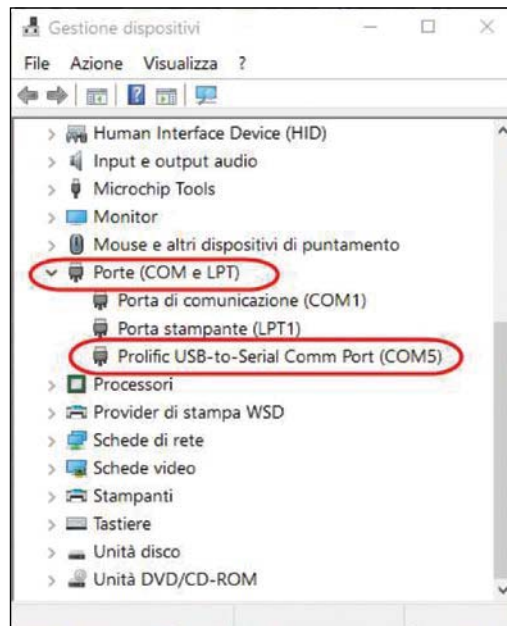


Fig. 4
Gestione dispositivi.

Prima di eseguire lo script si installano alcuni pacchetti necessari per la comunicazione seriale:

```
sudo apt-get install -y python-serial
sudo apt-get install -y python3-serial
sudo apt-get install -y python3-serial-asyncio
```

Notate che alcuni pacchetti potrebbero essere già presenti se si utilizza una versione di sistema operativo recente.

Lo script visualizza semplicemente i caratteri ricevuti e rimane in attesa dei caratteri da trasmettere; la relativa trasmissione avviene premendo il tasto Invio.

UTILIZZO DELLO SCRIPT DI ESEMPIO SULLA RPI3B+

Per utilizzare lo script di esempio su una scheda Raspberry Pi 3B+, dopo avere configurato la porta seriale ttyAMA0 attraverso il comando:

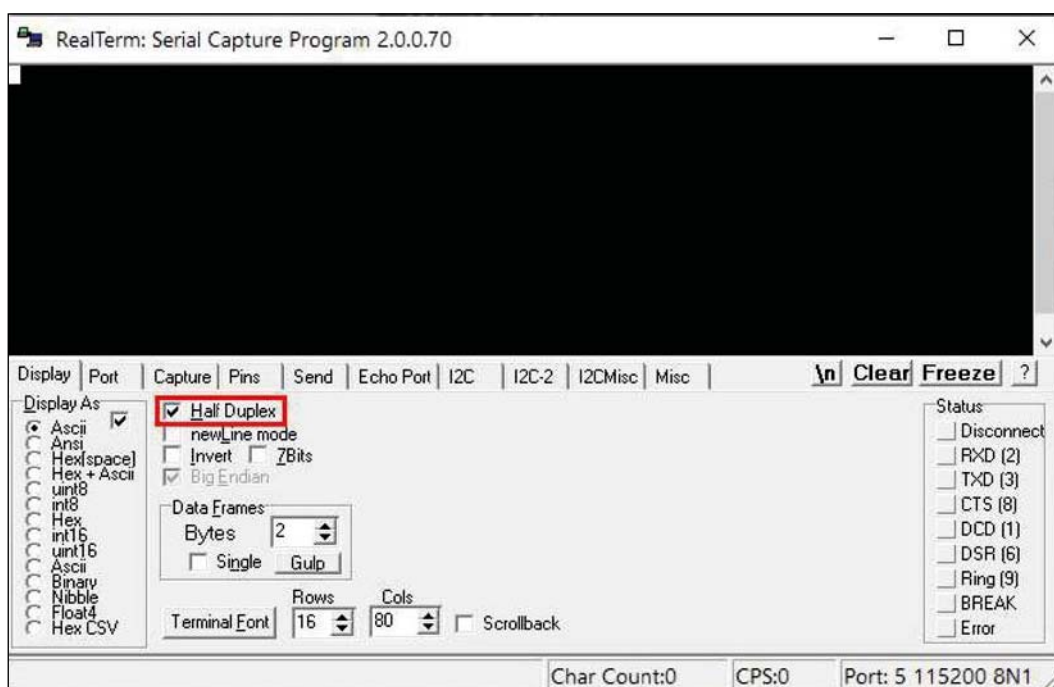
```
raspi-config
```

disattivando l'utilizzo della stessa da parte della console di sistema (operazione che si deve fare per tutti i modelli di Raspberry Pi al fine di disimpegnare la porta seriale dalla shell di sistema alla partenza) si deve anche disattivare il modulo Bluetooth di bordo, andando a modificare il file **/boot/config.txt** aggiungendo al fondo dello stesso la seguente riga:

```
dtoverlay=pi3-disable-bt
```



Fig. 5
Selezione
visualizzazione
Ascii e modalità
Half Duplex.



Fatto ciò si salva il file e si esce dall'editor.
In seguito si disattiva il servizio *systemd*, che in-
izializza il modulo Bluetooth (che essendo stato di-
sattivato con il comando precedente non verrebbe
trovato e la cosa causerebbe un errore); il comando
da impartire per disattivare tale servizio è:

```
sudo systemctl disable hciuart
```

per rendere operative le modifiche si riavvia la
Raspberry Pi con il comando:

```
sudo reboot
```

Esecuzione dello script di esempio per la comu-
nicazione seriale. Per eseguire lo script utilizzan-
do l'interprete Python2.x si utilizza il seguente
comando:

```
python ex-serial.py
```

A video apparirà la scritta Starting RXThread e poi
TX => seguito dal cursore verde.

Per utilizzare lo script di esempio con l'interprete
Python3.x si utilizza uno script leggermente mo-
dificato per poter essere correttamente eseguito
anche con le versioni di Python 3.x:

```
python3 ex-serial-py3.py
```

Nel corso di questa analisi si utilizzerà lo script per
Python2.x, fermo restando che quanto illustra-
to vale anche per lo script relativo all'interprete
Python3.x; le uniche differenze sono solo a livello
di codice (associate essenzialmente al fatto che
certi aspetti sono stati modificati nel passaggio
dalla versione 2.x alle 3.x e successive) ma non di
funzionalità. Lo script gestisce un semplice pro-
tocollo di comunicazione basato solo su caratteri
ASCII. Tutte le informazioni ricevute devono iniziare
con il carattere **\$** e terminare con il carattere **%**,
alla ricezione del terminatore viene visualizzata la
stringa di caratteri appena ricevuta; per la trasmiss-
ione è invece sufficiente digitare nel terminale i
caratteri desiderati e premere il tasto **Invio**.
L'implementazione del ciclo di ricezione è realizza-
ta in modo non bloccante utilizzando un approccio
che prevede l'utilizzo di un thread separato dal
corpo dell'applicazione principale.

Alla partenza, lo script trasmette un messaggio
di avvio **"Serial port running..."**, quindi se tutto è
stato configurato correttamente, lo stesso verrà
visualizzato nel terminale in esecuzione sullo
schermo del Personal Computer (nella **Fig. 6**
vedete la tipica rappresentazione). Scrivendo una
stringa di testo, ad esempio **\$CIAO%**, e si preme il
tasto invio si attiva la trasmissione dei caratteri
verso la Raspberry Pi, che provvede a visualizzarla.
Viceversa è possibile scrivere un testo nella fine-

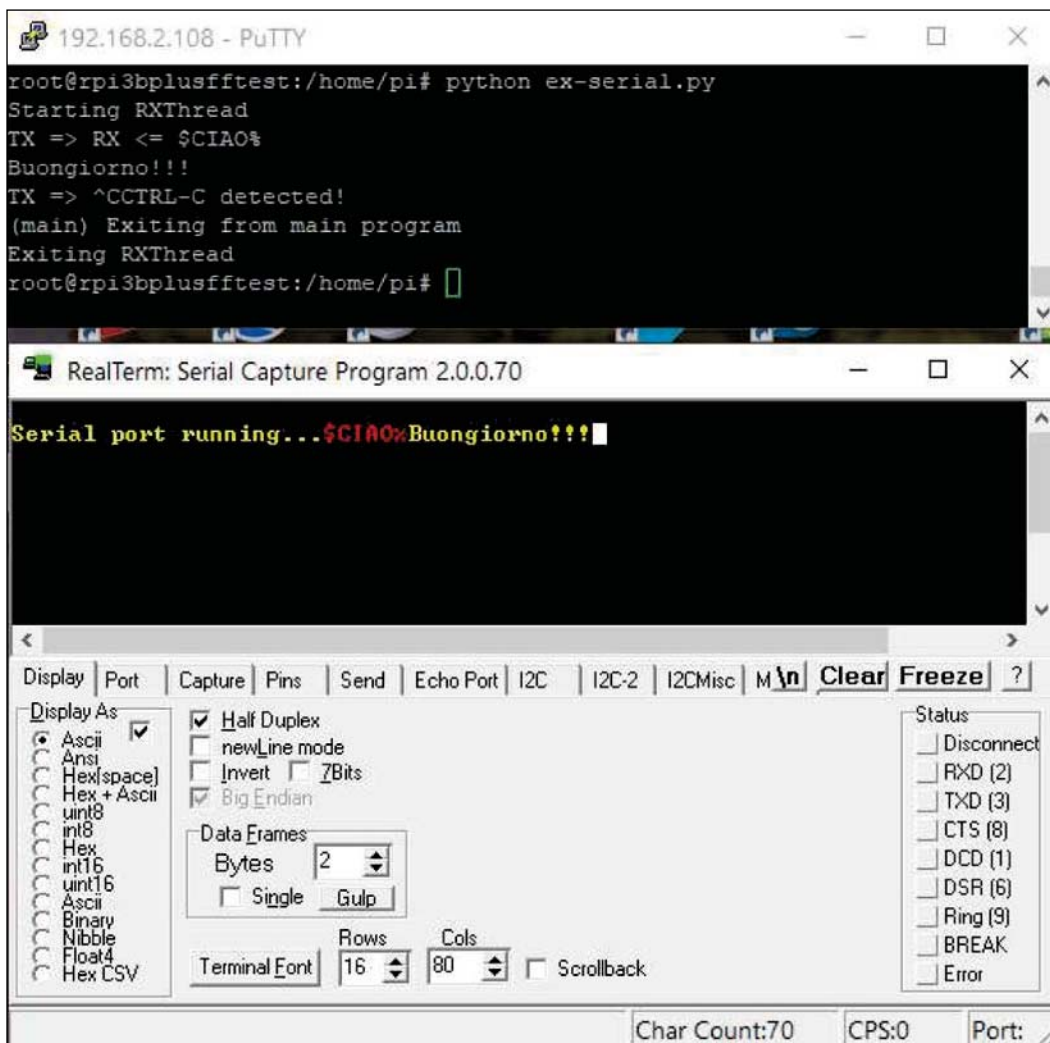


Fig. 6
Esempio di
esecuzione
dello script.

stra della shell sulla Raspberry Pi (in questo caso è sufficiente una qualsiasi sequenza di caratteri) seguita dalla pressione del tasto Invio per attivare la trasmissione verso il Personal Computer. Per terminare l'esecuzione dello script è sufficiente premere la combinazione di tasti CTRL+C (^C), che è quella che normalmente arresta l'esecuzione dei file eseguibili.

CONCLUSIONI

Bene, con questo termina la trattazione della nostra scheda UPS dedicata alla Raspberry Pi, che permette di implementare sia il mantenimento in esercizio in assenza della tensione di alimentazione principale (per il tempo consentito dalla batteria) sia l'arresto dei programmi in esecuzione e lo spegnimento controllato delle nostre schede. Non ci resta che augurarvi buon lavoro e buona

sperimentazione con la MiniUPS, se volete lavorare un po' sugli script di esempio che vi abbiamo descritto, per svilupparne di vostri e implementare specifiche applicazioni.



Cosa occorre?

Il miniUPS presentato in questo articolo (cod. MINIUPSRPI) è in vendita a Euro 58,00, la batteria LiPo da 420 mAh con controllo temperatura (cod. PLIB430) è disponibile a Euro 12,50. 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>**

Affidabilità e precisione nel palmo di una mano!

Sei alla ricerca di uno strumento di misura portatile, affidabile e preciso? Da noi trovi quello che fa per te! Scopri l'oscilloscopio palmare HANTEK, adatto per tutte le esigenze dal laboratorio professionale ai servizi di assistenza tecnica, all'impiego hobbistico, ideale quando la praticità è un requisito essenziale!



Istruzioni in italiano!

**Oscilloscopio portatile
2 canali - 40 MHz
con multimetro**

Hantek

Oscilloscopio digitale portatile a due canali con multimetro. Dispone di ampio display LCD TFT a colori da 2,8 pollici con una risoluzione di 320x240 pixel, batterie ricaricabili integrate, funzione di auto-calibrazione. 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. Alimentazione tramite batterie oppure adattatore di rete.



AC voltage
 DC voltage
 AC current
 DC current

Resistance
 Diode
 Capacitance
 Buzzer
 USB transfer
 Frequency
 Backlight

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



**cod. MH02D72GEN
€ 268,00**



**cod. MH02C42
€ 178,00**

FATTORE DI SMORZAMENTO DEL FINALE AUDIO

dell' Ing. MIRCO SEGATELLO



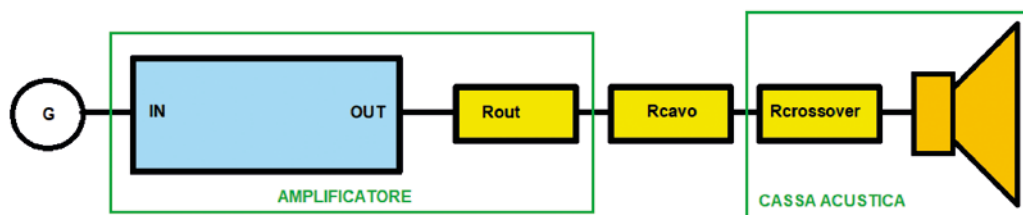
Scopriamo cos'è il fattore di smorzamento di un amplificatore audio e misuriamone l'entità con la scheda audio di un computer.

S

ono diverse le misure che si possono eseguire su un amplificatore audio, tra le più importanti riguardano la massima potenza erogabile su un determinato carico e a una frequenza di riferimento, la banda passante, la distorsione armonica (talvolta quella da intermodulazione)

ed il rapporto segnale/rumore (S/N ratio). Un'altra misura sicuramente importante, della quale peraltro ci occupiamo in questo articolo, è quella del fattore di smorzamento, un parametro che è direttamente correlato al valore della resistenza di uscita. Mentre un amplificatore ideale presenta una resistenza di uscita nulla per qualsiasi frequenza, nel caso reale una certa resistenza, seppur piccola, è sempre presente. Quest'ultima si somma alla resistenza dei cavi di collegamento (interni ed esterni alla cassa acustica) e alla resistenza

→ **Fig. 1**
Rappresentazione
di un tipico sistema
audio.



del filtro crossover, influenzando il controllo degli altoparlanti da parte dell'amplificatore, oltre che la potenza elettrica trasferita. La presenza di queste resistenze (si veda la **Fig. 1**) incide sulla qualità sonora in diversi modi: il primo effetto, di facile intuizione e correlato alla Legge di Ohm, è che parte della tensione erogata dall'amplificatore non giunge all'altoparlante a causa delle cadute di tensione sulle resistenze che dipendono dalla corrente circolante, a sua volta dipendente dall'impedenza dell'altoparlante che non è costante al variare della frequenza.

Anche solo mezzo ohm di resistenza comporta un'attenuazione di 1 dB su un carico di 4 ohm ed essendo questo valore variabile a seconda della frequenza, lo sarà anche l'attenuazione. Queste resistenze aggiuntive hanno anche effetto sulla capacità dell'amplificatore di accelerare e frenare la membrana dell'altoparlante, che si comporta come una massa con una certa inerzia: da qui deriva il termine "fattore di smorzamento"; il suo valore è determinabile come rapporto tra l'impedenza nominale dell'altoparlante e l'impedenza totale della sorgente che pilota l'altoparlante.

In **Tabella 1** riportiamo i tipici valori di smorzamento riscontrabili negli amplificatori.

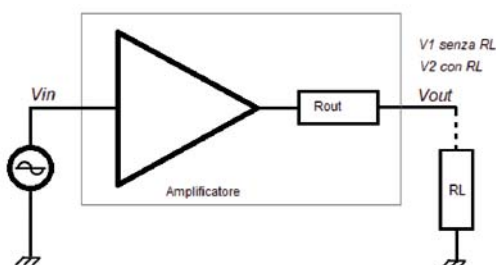
Il concetto di smorzamento si comprende considerando che l'altoparlante è elettricamente un sistema RLC, quindi un circuito risonante, che trova il suo analogo nella parte meccanica, rappresentata da una massa (quella dell'equipaggio mobile) da una capacità (l'elasticità e cedevolezza delle sospensioni) e da una resistenza, che è propriamente l'attrito; ebbene, la resistenza interna dell'amplificatore finale se è troppo alta smorza più lentamente i transienti e se è bassa li spegne prima. Tipico è il comportamento in risposta a segnali ad onda quadra, che rappresentano il test di stabilità per eccellenza.

Alto fattore di smorzamento significa, quindi, precisione nella riproduzione sonora, soprattutto di segnali ad alta dinamica e impulsivi, come ad esempio i suoni delle percussioni.

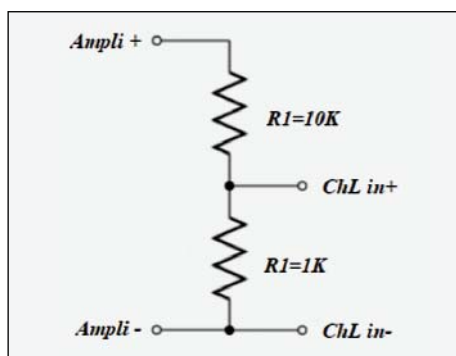
MISURA DEL FATTORE DI SMORZAMENTO

Per la misura del fattore di smorzamento è necessario sollecitare l'ingresso dell'amplificatore con un segnale sinusoidale alla frequenza di test e misu-

→ **Fig. 2**
Resistenza d'uscita e di carico.



→ **Fig. 3**
Collegamento
dell'attenuatore.



↓ **Tabella 1**
Fattore di smorzamento tipici degli amplificatori.

FATTORE SMORZAMENTO SU 8ohm	ROUT [ohm]	NOTE
<20	>0,4	Non accettabile
20-50	0,4-0,16	Ampli a valvole
50-100	0,16-0,08	Ampli a stato solido poco retroazionato
100-500	0,08-0,016	Ampli a stato solido mediamente retroazionato
>500	<0,016	Ampli a stato solido molto retroazionato

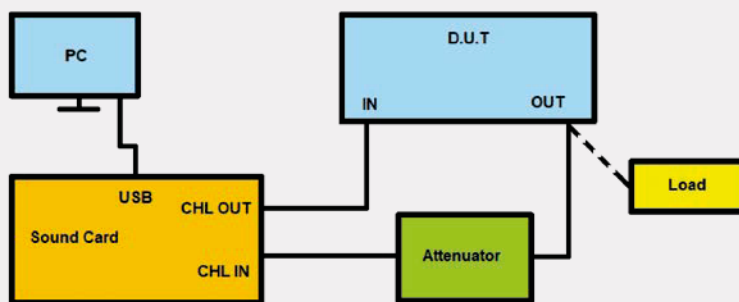


Fig. 4
Collegamenti hardware per la misura.

rare la corrispondente tensione in uscita a vuoto e con un carico noto, nel nostro caso una resistenza da 8 ohm che chiameremo R_L (come evidenziato in **Fig. 2**). Per ciascuna frequenza di misura avremo una tensione a vuoto V_1 ed una tensione a carico V_2 , dalle quali è possibile determinare sia il fattore di smorzamento (Dumping Factor, in inglese...) che il valore della resistenza di uscita, con le seguenti formule:

$$DF = \frac{V_2}{V_1 - V_2}$$

$$R_{out} = \frac{V_1 - V_2}{V_2} \cdot R_L$$

Per poter leggere la tensione in uscita con estrema risoluzione in un range di frequenze compreso tra 20 e 20 kHz, il classico multimetro non risulta adeguato e l'oscilloscopio è da scartare per via della bassa risoluzione nella misura della tensione, in quanto l'ADC interno è normalmente di soli 8 bit. La scelta ottimale è l'utilizzo di una scheda audio che, anche nelle versioni più economiche, non

scende sotto i 16bit di risoluzione; inoltre rende possibile gestire sia la generazione che l'acquisizione dei segnali.

La scheda audio dovrà produrre un segnale sinusoidale alla frequenza di prova e leggere il relativo segnale in uscita, che però risulta di ampiezza troppo elevata per l'ingresso della scheda audio, pertanto sarà necessario anteporre un attenuatore formato da un semplice partitore di tensione con due resistenze da 1K e 10K come in **Fig. 3**.

La misura è suddivisa in due fasi: una prima fase senza carico, come nello schema applicativo riportato in **Fig. 4**, ed una seconda fase in cui viene applicato un carico resistivo (R_L) del valore di 8 ohm. Non è possibile utilizzare come carico le casse acustiche, perché la loro impedenza è di tipo complesso (presenta una componente induttiva e capacitiva) e variabile con la frequenza; insomma, non è una semplice resistenza ma un'impedenza vera e propria.

Se la lettura della tensione in uscita verrà effettuata ai capi dei morsetti dell'amplificatore misurerete il fattore di smorzamento (e quindi R_{out}) del solo amplificatore, mentre se i collegamenti verranno

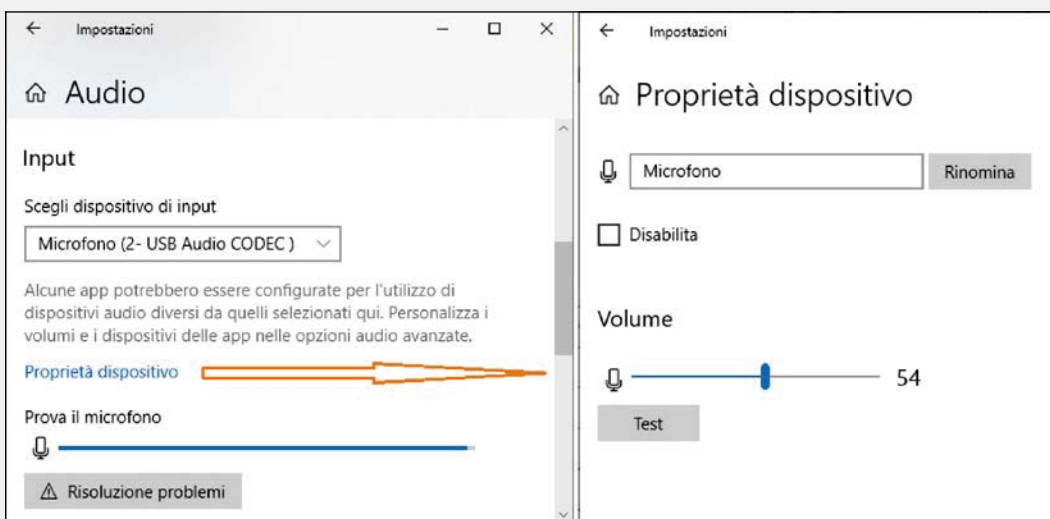


Fig. 5
Configurazione ingresso scheda audio.

→ **Fig. 6**
Configurazione
uscita scheda
audio.



effettuati a valle dei cavi di collegamento che normalmente usate per connettere le casse acustiche, avrete un'informazione più completa sulla qualità del vostro impianto di riproduzione. Non è possibile misurare anche la resistenza del crossover, se non manomettendo le casse acustiche.

CONFIGURAZIONE SCHEDA AUDIO

La scheda audio integrata nel PC va benissimo, ma se avete paura di danneggiarla a causa di un collegamento errato, potete acquistarne una esterna da connettere via USB al PC ad un costo irrisorio. Prima di procedere con la misura dovreste configurare opportunamente la vostra scheda audio; faremo riferimento all'utilizzo del sistema operativo Windows 10. Per quanto riguarda l'ingresso (microfono) è opportuno configurarlo per un basso guadagno per evitare che lo stadio di ingresso vada in saturazione, come si vede in **Fig. 5**. Per quanto riguarda l'uscita è invece importante sia la più elevata possibile, si veda la **Fig. 6**.

↓ **Fig. 7**
Schermata
iniziale del
software.

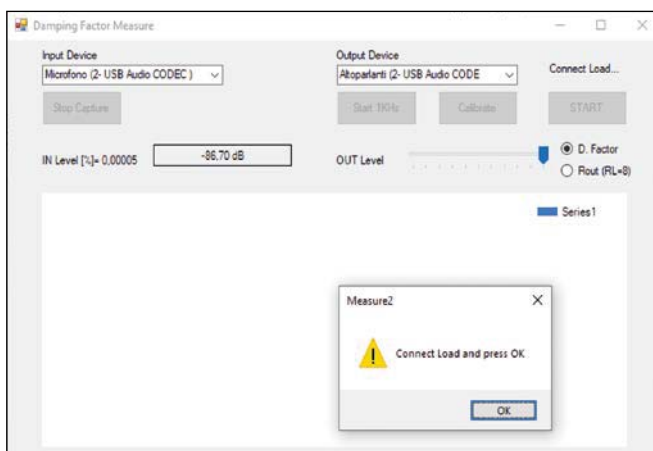
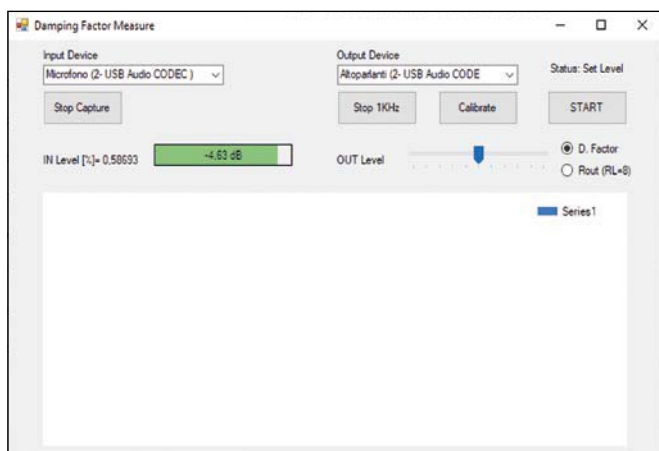
→ **Fig. 8**
Prima parte
della misura
conclusa.

IL SOFTWARE

Per eseguire la misura in modo automatico abbiamo sviluppato un apposito software sviluppato in C#. Dopo aver installato e configurato il software dovreste selezionare il vostro dispositivo di input e di output, abilitare la cattura e la generazione del segnale di prova per verificare il livello in ingresso, è ottimale quando si attesta attorno i -3dB, eventualmente regolate il livello di uscita dal software oppure dal vostro impianto. Selezionate sulla destra se volete misurare il fattore di smorzamento o la resistenza di uscita, verificate che tutti i collegamenti siano a posto, quindi avviate la misura (**Fig. 7**). In alto a destra verrà visualizzato lo stato di avanzamento con la frequenza di test; ed in pochi secondi la prima fase sarà conclusa. Terminata questa fase (**Fig. 8**) un messaggio vi avviserà di collegare il carico resistivo in uscita dall'amplificatore: confermate ciò con il pulsante OK, ancora pochi secondi e la misura sarà conclusa, se non ci sono stati errori comparirà un grafico con i valori misurati.

In **Fig. 9** il risultato di un tipico amplificatore a stato solido ad elevata retroazione; notate come il fattore di smorzamento tenda a diminuire con l'aumentare della frequenza e ciò indica che lo stadio di uscita ha una resistenza che aumenta per frequenze elevate. In genere è preferibile amplificatori con retroazione più contenuta ed un fattore di smorzamento più basso ma costante al variare della frequenza.

Ricordiamo, infine, che la misura è sensibile ai collegamenti di massa, pertanto è consigliabile utilizzare un portatile alimentato a batterie sconnesso dall'impianto elettrico di casa. La resistenza di uscita di un amplificatore oltre a variare con la frequenza dipende dal punto di lavoro dei transi-



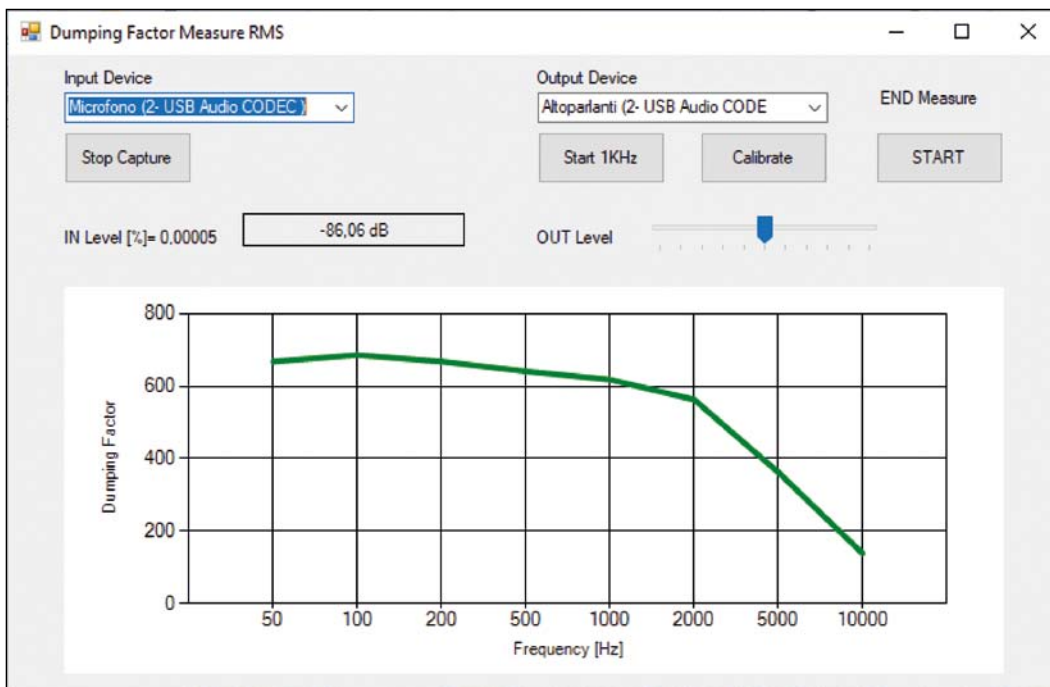


Fig. 9
Misura di DF
conclusa.

stor finali, pertanto potreste trovare valori diversi a seconda dell'ampiezza del segnale di test. Amplificatori valvolari a parte, un qualsiasi componente a stato solido presenta comunque un fattore di smorzamento sufficientemente elevato da non influire le prestazioni finali del sistema; semmai è il caso di curare attentamente il collegamento

tra l'amplificatore e le casse acustiche utilizzando buoni connettori e cavi di qualità, così come è importante che i collegamenti interni delle casse acustiche, compreso il crossover, siano di qualità. Nel nostro amplificatore usato come cavia la resistenza di uscita si mantiene sotto gli 0,02ohm fino a 2KHz, valore più che rassicurante. ■

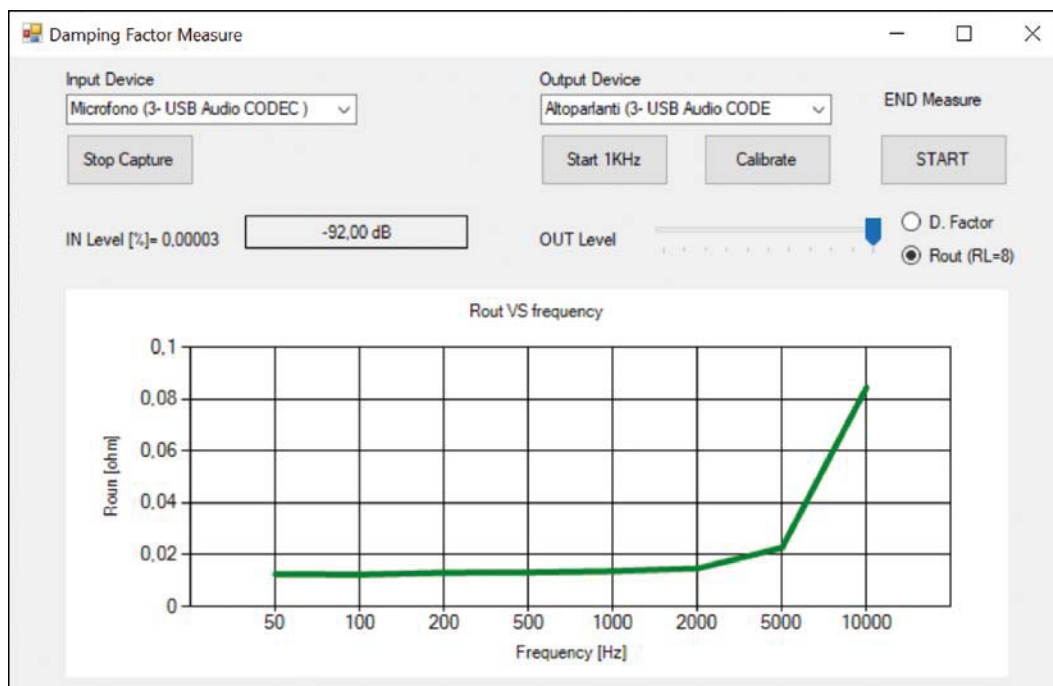


Fig. 10
Misura di Rout
conclusa.

Nuovo allarme evoluto WiFi-GSM/GPRS



€119,00

Cod. FR783



Controllabile con:



Smart Life



Tuya Smart



Amazon Alexa



Google Home

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

Prezzi IVA inclusa



Accessori
opzionali:

Sensore PIR
wireless



€14,00

Cod. FR783PIR

Sensore
porte/finestre
wireless



€11,00

Cod. FR783MAGN

Radiocomando
supplementare



€9,00

Cod. FR783RAD

Transponder
a portachiavi



€5,00

Cod. FR783TAG

Sirena wireless
da interno



€29,00

Cod. FR783SIR

Sirena wireless
da esterno con
pannello solare



€65,00

Cod. FR783SIREXT

ALLARME BASATO SU AI

dell'Ing. **DANIELE DENARO**



Sistema anti-intrusione sensibile solo alle figure umane, che genera notifiche di allarme tramite instant messaging su Telegram.

Q

ualche numero addietro vi abbiamo presentato un sistema di sviluppo piccolo e compatto, più o meno come Raspberry Pi, ma molto più adatto ad applicazioni che richiedono un utilizzo intensivo di calcoli come quelle basate sull'Intelligenza Artificiale e sulle reti neurali in modalità "deep-learning".

Questo mini super computer consiste nella scheda "Jetson Nano Developer Kit". L'hardware e la struttura software di base sono stati presentati nel numero 244 della nostra rivista, però ricordiamo brevemente le sue caratteristiche:

- processore quadri-core ARM Cortex-A57 con clock a 1,5 GHz;
- GPU NVIDIA CUDA con 128 core;
- RAM 4 GB (ma recentemente è stata presentata anche la versione a 2 GB);

- Socket per scheda microSD;
- connettore di espansione header 40 pin e connettore per webcam ambedue compatibili con Raspberry Pi;
- HDMI, Display-Port, 4 USB tipo 3, Ethernet.

Ripariamo ora di questo hardware (**Fig. 1**) perché vogliamo continuare ad affrontare tematiche di Intelligenza Artificiale e per poterlo fare non possiamo che utilizzare sistemi con una sufficiente capacità computazionale. Per cui, a parte un buon PC con adeguata scheda grafica CUDA, non rimane che considerare hardware di sviluppo specializzato ma di costo contenuto come Jetson.

Prima di affrontare l'applicazione suggerita dal titolo è il caso di farvi presente che il repository GitHub della nostra rivista, dedicato al software IA è stato completamente rivisto nell'ottica di farlo diventare un contenitore del software per diverse strutture ed applicazioni Intelligenza Artificiale. Il repository è accessibile da https://github.com/open-electronics/Artificial_Intelligence.

Purtroppo il software a corredo di Jetson non ha le stesse caratteristiche di semplicità e completezza di quello per Raspberry Pi. Per esempio, OpenCv, compreso nella distribuzione del sistema operativo, non è compilato per l'utilizzo delle caratteristiche CUDA! Ne consegue che per utilizzarlo si è dovuto ricompilarlo a partire dai file sorgente; ma attenzione, la compilazione comprende diversi parametri e richiede anche alcune ore! L'obiettivo del sito GitHub sarà anche quello di cercare di mettere a disposizione dell'utente software già operativo, per cui sarà caricata anche la versione compilata di OpenCV con CUDA.



Fig. 1
La board
Jetson Nano.

Anche una distribuzione del sistema operativo Linux per Jetson che comprenda già alcuni software operativi ed utili, sarà caricata sul sito, anche se con qualche difficoltà a causa della dimensione di circa 12 GB (infatti Github non è pensato per grossi file binari, per cui il sito contiene un link per il downloading che avverrà automaticamente da un altro server). In generale si potrebbe dire che l'attenzione di NVIDIA è rivolta più all'utilizzo della IA distribuita e professionale che a quella hobbistica. Da qui l'aspetto molto tecnico e poco semplificato del software a corredo anche se l'utilizzo come computer (con tastiera e schermo) rimane immediato e classico.

Anche il desktop remoto ha difficoltà ad essere implementato; mentre non c'è nessun problema per l'utilizzo da remoto con sessioni SSH anche con supporto Xserver (per esempio utilizzando su PC il software MobaXterm).

SISTEMA DI ALLARME CON INTELLIGENZA ARTIFICIALE

I sistemi di allarme comunemente basati su una telecamera, riescono a rilevare il movimento ma non possono distinguere quale è stata l'origine del movimento: un essere umano, un gatto, un cane, un uccello o perfino una foglia possono scatenare l'allarme. Questo è sicuramente vero per le telecamere poste all'esterno, ma può succedere anche all'interno di abitazioni, se si è lasciata una finestra aperta o sono presenti animali da compagnia nell'appartamento. Ovviare con un aggiustamento della soglia di intervento non risolve veramente il problema. In genere queste soglie si basano sulla grandezza dell'area interessata, ma una farfalla vicino all'obiettivo può scatenare l'allarme. Però oggi esistono già sistemi software per il riconoscimento di oggetti. Per esempio la rete deep-learning YOLO (You Only Look Once) è abbastanza efficiente e veloce in questo compito. Chiaramente la velocità dipende dall'hardware utilizzato ma esiste comunque una versione "tiny" (leggera) della rete. Le differenze di struttura e dimensione tra le due versioni sono notevoli, come anche le rispettive velocità di esecuzione e affidabilità. Nel caso di applicazione in un allarme, i tempi di latenza nel riconoscimento dell'oggetto "pericoloso" non sono così essenziali, a patto che si mantengano entro il secondo.

Al contrario se volessimo utilizzare questo tipo di rete neurale per il pilotaggio di un "rover" avremmo bisogno di un "frame rate" di almeno 10/20 frame/secondo. La versione completa della rete è

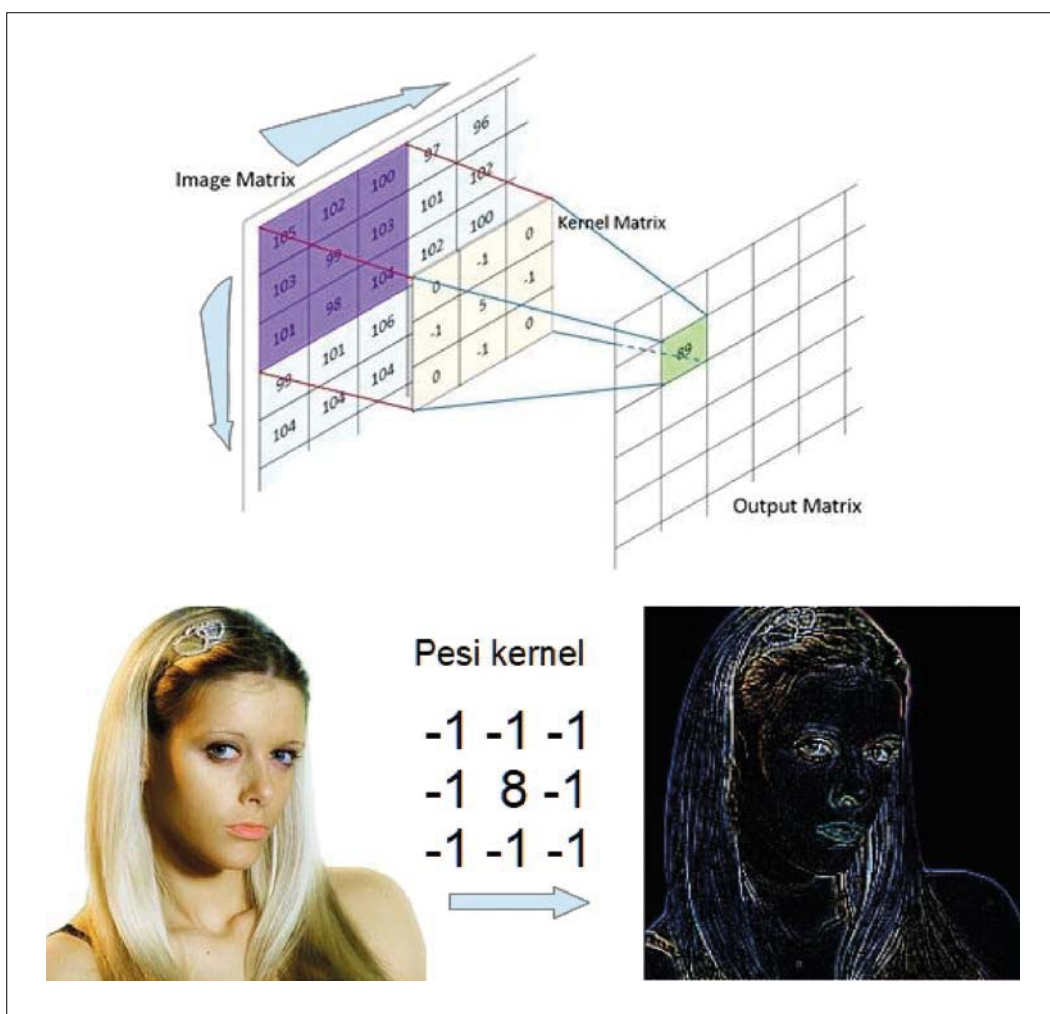


Fig. 2
Esempio e
schematizzazione
di un filtro
convolutivo.

sicuramente più affidabile nel riconoscimento degli oggetti e, quindi in linea di massima, la privilegeremo, anche se sarà sempre possibile passare dall'una all'altra con un semplice aggiornamento di un parametro.

Tenete presente che la rete può discriminare un'ottantina di oggetti ed esseri differenti come: persona, cane, gatto, uccello, mucca, pecora, orso, ecc. L'elenco completo lo potete trovare sul web all'indirizzo https://github.com/open-electronics/Artificial_Intelligence/tree/master/Deep_Learning/YoloPython/README.md.

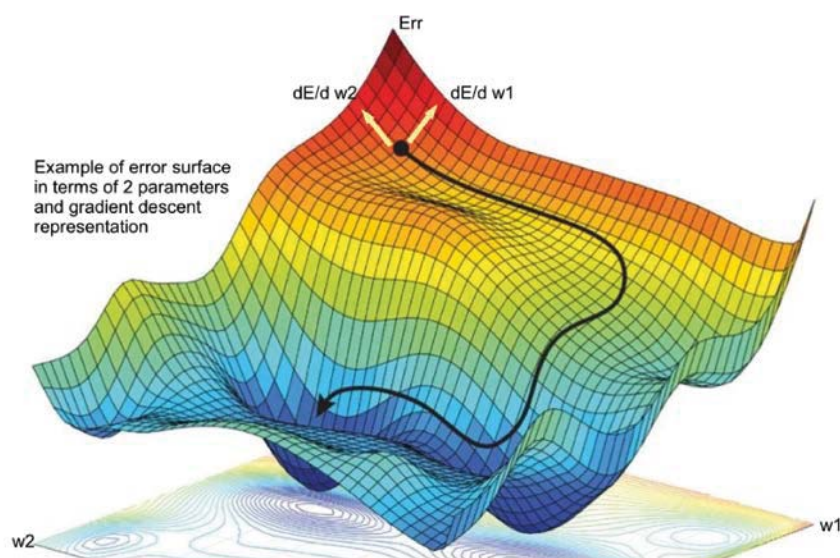
Quindi è possibile adattare questo sistema ad altri usi come per esempio una trappola fotografica. Come sempre, gli articoli dedicati a progetti di Intelligenza Artificiale, hanno lo scopo di suggerire ed illustrare le strutture e le potenzialità delle reti neurali e *machine learning*, piuttosto che creare progetti industriali. Per cui ogni suggerimento dei lettori sarà più che benvenuto.

Prima di passare al progetto vero e proprio, illustreremo brevemente le caratteristiche del deep-learning per l'analisi di immagini ed in particolare la struttura YOLO, di cui abbiamo già in effetti parlato nel numero 242.

YOLO fa parte delle reti convolutive, che sono le strutture deep-learning utilizzate per analizzare le immagini. Una rete convolutiva non fa altro che applicare filtri convolutivi all'immagine in modo da isolare pattern che possano essere significativi ai fini del risultato che si vuole ottenere. Poiché stiamo parlando di reti neurali è ovvio che si tratta di un processo di addestramento e non di programmazione. Ovvero questi filtri vengono addestrati ad evidenziare pattern che saranno utili a fornire una risposta finale più simile possibile agli esempi di addestramento.

Un filtro convolutivo (**Fig. 2**) è sostanzialmente una somma pesata dei pixel che circondano il pixel dato, ed è applicata ad ogni pixel dell'immagine. I pesi

Fig. 3
Discesa del
gradiente.



del filtro sono i parametri che vengono addestrati durante la presentazione degli esempi di addestramento (con il sistema "Error Back Propagation": ovvero discesa del gradiente schematizzato in **Fig. 3**).

Purtroppo i filtri sono numerosi, lavorano in parallelo e sono anche a strati successivi (**Fig. 4**).

Ne consegue che la struttura può essere veramente pesante. Si parla di decine o di centinaia di MB di parametri.

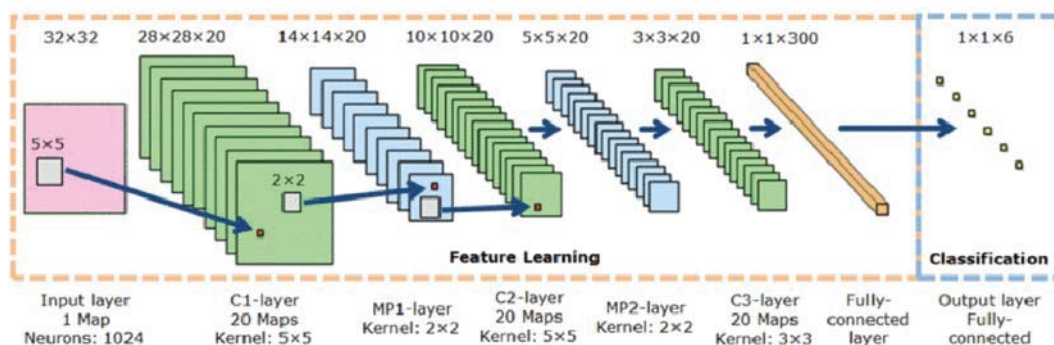
La struttura YOLO non solo discrimina un certo numero di oggetti (un'ottantina se addestrata con immagini definite dalla collezione chiamata "COCO"), ma li localizza anche, suddividendo l'immagine in regioni. Sono state pubblicate diverse versioni del sistema di cui la più recente è la versione YOLOv4. La struttura YOLOv4 ha 53 strati convolutivi, più numerosi strati successivi per la composizione dei pattern rilevati e per la localizzazione degli oggetti all'interno della immagine; per un totale di circa 250 MB di parametri.

La struttura semplificata, YOLOv4-tiny, ha invece,

per esempio, solo 9 strati convolutivi per un totale di circa 25 MB di parametri.

La struttura YOLO è spesso associata al software di esecuzione DARKNET in C, perché è stato il software originario prodotto dallo stesso autore della struttura YOLO (Joseph Redmon, nick: pjreddie). In realtà noi useremo la rete utilizzando il framework OpenCv che ha un modulo per il caricamento e l'esecuzione di diversi tipi di strutture di reti deep-learning (modulo "dnn"). Utilizzeremo OpenCv sia perché nasce per l'elaborazione di immagini e quindi ha diverse funzioni per l'acquisizione e il pre-trattamento di esse, ma anche soprattutto, perché può essere utilizzato in un programma Python. Utilizzare il linguaggio Python permette di pubblicare e modificare i programmi facilmente essendo un linguaggio interpretato. Se siete preoccupati per le performance, state tranquilli per due motivi: Python ha in realtà delle funzioni compilate che vengono chiamate dall'interprete (processo di binding), il peso dell'elaborazione di queste strutture deep-learning è

Fig. 4
Esempio di
deep learning
convolutivo.



molto maggiore dell'eventuale *overloaded* dovuto all'interprete. In effetti le prove effettuate con DARKNET e con Python-OpenCv hanno evidenziato quanto sia più importante la compilazione e installazione di OpenCv rispetto all'utilizzo di Python. Volendo utilizzare questo sistema su Jetson, era importante che OpenCv sfruttasse appieno l'elaborazione distribuita dell'architettura CUDA. Purtroppo, come si è detto, OpenCv incluso nel software di base di Jetson non risultava compilato con l'opzione CUDA per il modulo "dnn", per cui si è provveduto a ricompilare OpenCv con l'opzione CUDA ed il multi-processing, visto che Jetson ha anche quattro processori. Per chi volesse cimentarsi nel processo di compilazione, conviene installare prima Cmake (o meglio la sua versione grafica : *cmake-gui*), se non già presente, caricare la "CmakeLists.txt" e controllare tutti i parametri ed in particolare CUDA e MP, infine dotarsi di molta pazienza visto che occorre qualche ora. In ogni caso sul sito sarà caricato un completo ambiente Jetson da scaricare sulla schedina SD.

FUNZIONALITÀ DEL SISTEMA DI ALLARME.

Il sistema di allarme è pensato per essere utilizzato sia all'interno di ambienti sia soprattutto all'esterno dove il discrimine della figura umana è più importante (la **Fig. 5** espone come il sistema percepisce la scena). Naturalmente se posizionato all'esterno il contenitore deve essere più protettivo. Il collegamento con internet tramite un router può

avvenire tramite WiFi ma in ogni caso il sistema deve essere costantemente alimentato a causa del suo assorbimento. L'utilizzo di una batteria non è pensabile, a meno di attivarlo di volta in volta tramite un sensore di movimento. Ma in questo caso si perderebbe l'immediatezza del riconoscimento e soprattutto i numerosi riavvii, a causa di falsi allarmi, consumerebbero velocemente la batteria. Benché il sistema possa essere ovviamente dotato dell'attivazione di una sirena, esso è pensato per avvertire remotamente l'utente della possibile intrusione. Il modo più immediato per allarmare l'utente è quello di sfruttare l'onnipresente telefonino. Per evitare di dotare il sistema di una propria scheda "sim", è possibile utilizzare internet per inviare un messaggio a se stessi, su WhatsApp o Telegram. WhatsApp permette di inviare un messaggio a se stessi non in maniera ufficiale, ma tramite un trucco: basta creare un gruppo e poi cancellare tutti i partecipanti. In questo modo il gruppo riceve il messaggio che il suo creatore vede in esclusiva. Ma c'è un problema: il messaggio arriva senza notifica sonora perché WhatsApp rileva che è dello stesso proprietario. Per un allarme non è il massimo! Passando a Telegram (una chat simile a WhatsApp ed anche più sicura), il problema del messaggio a se stessi è risolto automaticamente perché Telegram lo consente, ma rimane il problema della mancata notifica sonora. Si è pensato, allora di sfruttare una caratteristica di Telegram non presente in WhatsApp. In Telegram è possi-

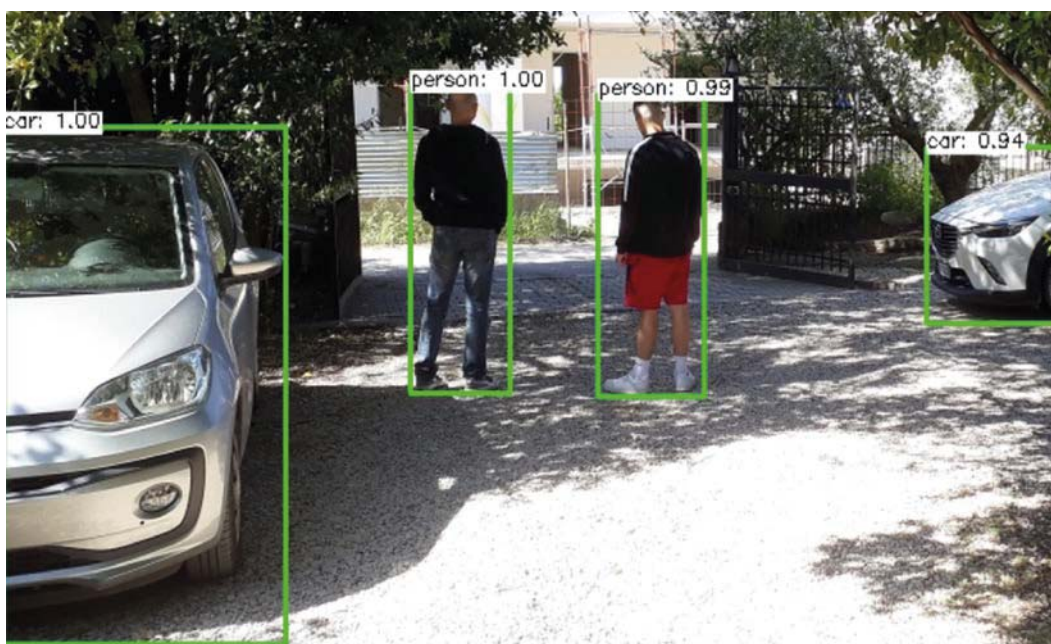
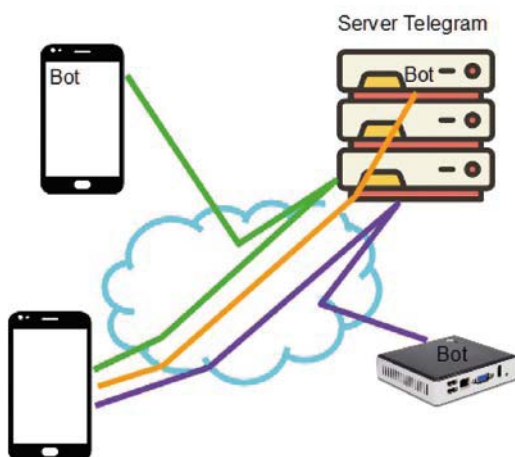


Fig. 5
Esempio
reale di come
il sistema
vede la
scena.

Fig. 6
Messaggi a
Bot Telegram
diversamente
istallati.



bile creare dei Bot. Un Bot è sostanzialmente un utente fittizio ed automatico. Nel senso che può ricevere e spedire messaggi ma non è collegato ad un numero telefonico (l'indirizzo è il suo nickname; anche se è collegato al suo creatore dal punto di vista della proprietà). Alcuni Bot sono pubblici, nel senso che vengono elencati in liste pubbliche, ed altri privati, ma devono essere collocati su computer collegati in rete (anche piccoli computer come Raspberry Pi) oppure risiedere sullo stesso telefonino (**Fig. 6**) Telegram mette a disposizione delle API in vari linguaggi per creare questi risponditori automatici.

Un Bot deve comunque avere un proprietario ed essere autorizzato tramite codici forniti da Telegram. L'idea è quella di scrivere un Bot in Python su Android in modo da lanciare un suono mp3 sul telefonino al ricevimento dell'allarme. La scelta di non utilizzare Bot già disponibili è dovuta a due fattori: un Bot semplice per questo scopo e che risieda sul telefonino non è disponibile, e installare un Bot di gestione remota del telefonino o utilizza-

re una App per la creazione di un Bot Android non è sicuro e consigliabile, perché potrebbero asservire il vostro telefono. Uno script Python è invece leggibile e verificabile. I Bot, pubblicamente disponibili, in genere risiedono sul server Telegram o su altri server e funzionano come fornitori di servizi intelligenti preoccupandosi di rispondere fornendo informazioni specializzate quando interrogati.

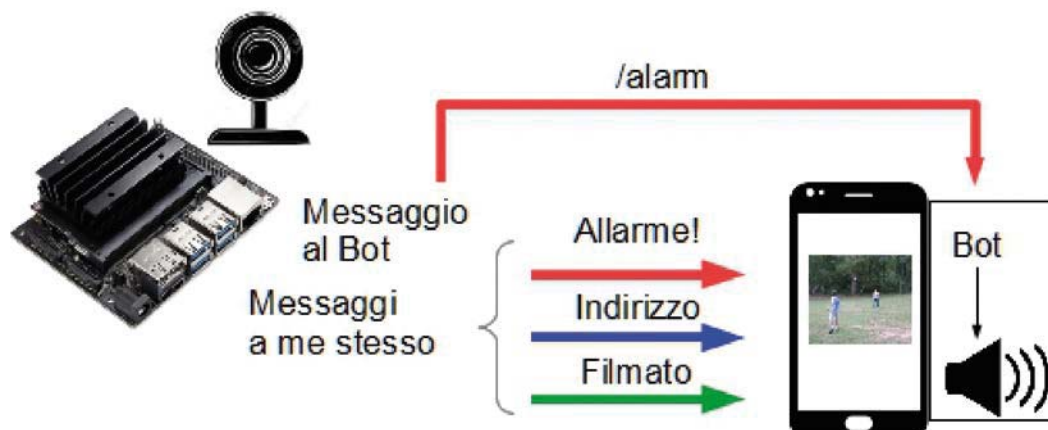
Ma noi abbiamo bisogno di qualcosa di diverso: un Bot sul telefono che reagisca ad un comando remoto. Telegram verrà utilizzato quindi in due modi:

- installando le sue API Python su Jetson per simulare un utente e spedire messaggi di allerta a se stessi;
- installando le sue API Python su smartphone Android per creare un Bot che attivi la suoneria.

In ogni caso queste API faranno riferimento, tramite codici di sicurezza, all'utente Telegram di riferimento che ha provveduto a registrarsi come tale. Per questo motivo, dopo essersi registrati sul sito Telegram (se non si è già suoi utenti), tramite App o anche da computer, bisogna :

1. Andare sul sito <https://my.telegram.org/> e dopo essersi autenticati tramite il numero telefonico andare alla voce API e chiedere di creare un'applicazione; in questo modo vi verranno forniti due importanti codici segreti:
 - **Application Id**: codice numerico;
 - **Codice crittografico** (hash): lungo codice alfanumerico (*case sensitive*).
2. Azionare da telefono (o da client web) l'icona "BotFather", sempre presente nella pagina chat. BotFather (a sua volta un Bot standard) vi consente di creare un nuovo Bot inviando il messaggio `/newbot`. In realtà a BotFather potete inviare diversi comandi e per averne un elenco usate il comando `/help`. Dopo il coman-

Fig. 7
Sistema di allarme.



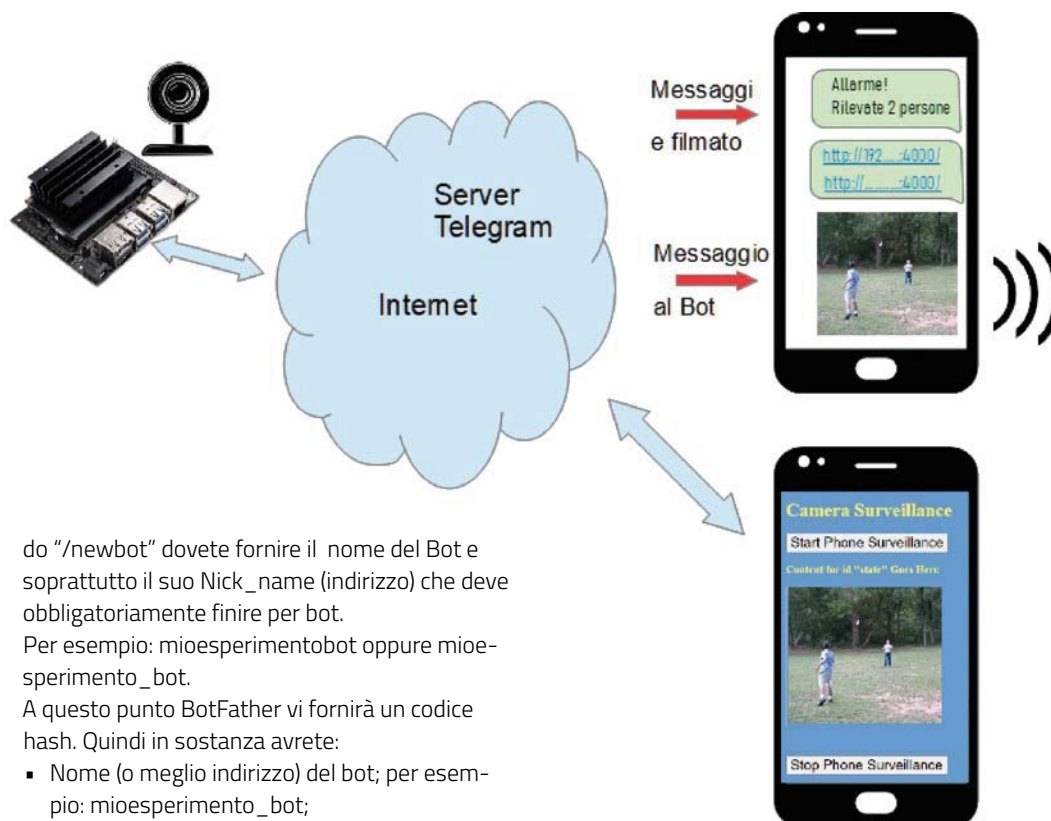


Fig. 8
Sistema di sorveglianza e allarme.

do `/newbot` dovete fornire il nome del Bot e soprattutto il suo Nick_name (indirizzo) che deve obbligatoriamente finire per bot.
Per esempio: `mioesperimentobot` oppure `mioesperimento_bot`.

A questo punto BotFather vi fornirà un codice hash. Quindi in sostanza avrete:

- Nome (o meglio indirizzo) del bot; per esempio: `mioesperimento_bot`;
- Codice crittografico (hash): (diverso per ogni bot e diverso da quello al punto 1).

Il punto 1 permette di poter inviare messaggi da programma come utenti Telegram, ovvero identificati dalla vostra utenza telefonica. Mentre il punto 2 permette di creare un Bot di cui voi siete responsabili e che risponde all'indirizzo `@mioesperimento_bot`. In realtà manca ancora un passaggio. L'applicazione Python che gira su Jetson deve essere autorizzata ulteriormente la prima volta. Ovvero è necessario un "Sign-in" specialistico per il sistema da cui parte il collegamento come "client" Telegram. Questo può essere fatto dallo script di utilità Python "TelegramSignIn.py", che chiede conferma del numero telefonico dell'utenza ufficiale Telegram (numero che deve partire con il prefisso nazionale +39), a cui segue un messaggio Telegram su telefonino (o su client WEB) in cui viene fornito un codice (numerico di 5 cifre) che va digitato nella applicazione Python che così completerà il riconoscimento. L'applicazione crea, infatti, un file "TelAlarm.session" nella directory della applicazione; file che Jetson utilizzerà ad ogni connessione.

Per gestire i collegamenti Telegram e creare il Bot sullo smartphone, si è preferito installare la libreria Python "Telethon". Telegram può essere gestito

con un protocollo http, ma per semplificare questo aspetto del progetto, Telethon garantisce una certa astrazione.

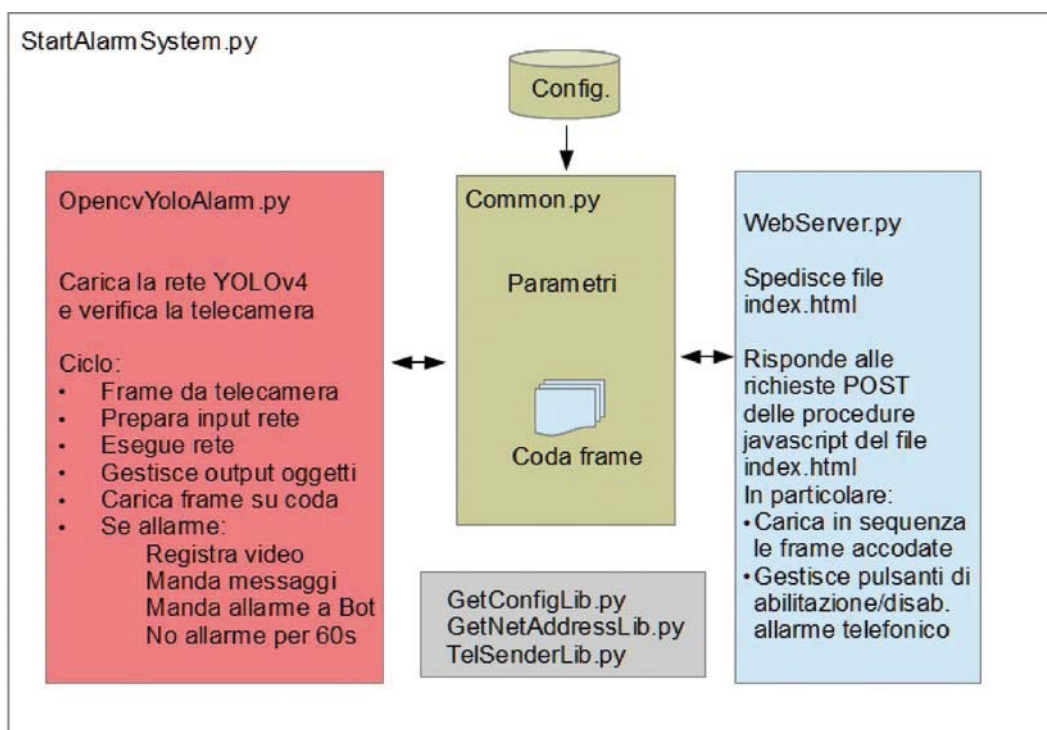
La documentazione la potete trovare sul web alla pagina <https://docs.telethon.dev/en/latest/>.

Quindi ricapitolando: il sistema controlla costantemente la scena rilevando la presenza di diversi oggetti; quando tra questi ci sono una o più persone, attiva l'allarme (che può essere collegato ad una sirena). L'allarme scatena una serie di eventi (**Fig. 7 e Fig. 8**):

- vengono registrati (in formato mp4) 10 secondi di filmato (che sovrascrive il precedente);
- ci viene inviato un messaggio in cui si allegano anche gli indirizzi locale e pubblico del sistema Jetson;
- viene auto-inviato il filmato registrato;
- viene inviato un messaggio di allarme al Bot (residente sullo smartphone) che attiva un suono predisposto.

Il sistema si pone in pausa per 60 secondi (modificabili), per evitare messaggi consecutivi. Il sistema inoltre gestisce un elementare web-server per permettere di monitorare l'ambiente in qualunque momento.

Fig. 9
Schema a blocchi
del software.



L'HARDWARE DEL PROGETTO

Il sistema hardware è composto semplicemente dalla scheda "Developer Kit" di Jetson Nano corredata dell'alimentatore (da 4 Ampere), della ventola a velocità controllata di Jetson posta sul dissipatore (o in alternativa una ventola sul contenitore), dell'adattatore WiFi (per esempio una chiavetta USB-WIFI come EDUP NANO con N8508), ed infine una telecamera. Quest'ultima può essere una "Picamera" ovvero una webcam per Raspberry Pi ma solo versione 2 con chip IMX219 (mentre quella con OV5647 non è supportata). Oppure una semplice webcam USB (meglio controllare la compatibilità sul sito Jetson, ma in genere quelle standard di buona qualità vanno bene). Il tutto va inserito in un contenitore che dovrà proteggere dall'acqua il sistema se esso va posto all'esterno. In ogni caso è importante il raffreddamento perché il sistema scalda molto quando è sollecitato dai calcoli di una complessa rete neurale. Evidenziamo alcuni consumi rilevati:

- Accensione del sistema: circa 0.5 A
- Sistema in attesa di input interattivo: circa 0.2 A
- YOLOv4-Tiny : circa 1.2 A
- YOLOv4 : circa 1.6 A
- Sistema in attività di sviluppo con IDE pesante come PyCharm : circa 0.8 A
- Attività Wi-Fi con chiavetta : circa 0.15 A come contributo alle altre attività

Ne consegue che in modalità sorveglianza il sistema consumerà circa 8W se con YOLOv4 o circa 6W se con la versione più leggera.

SOFTWARE DEL SISTEMA

Il sistema software è sviluppato tutto in Python (con l'appoggio di OpenCv) ed è composto sostanzialmente da due moduli (**Fig. 9**):

- Programma di caricamento ed esecuzione della struttura deep-learning YOLO, con controllo degli oggetti rilevati ed eventuale allarme con la complicità della libreria (Python Telethon) per Telegram
- Web server che riceve le frame dal programma precedente e le mette a disposizione di un eventuale browser.

Un programma di start ("StartAlarmSystem.py") si occupa di leggere eventuali parametri di configurazione e di lanciare i due moduli come thread. Un file di configurazione ("alarmsystem.cfg") definisce numerosi parametri fra cui soprattutto in vari codici per Telegram ed il Bot. Poiché il Bot di allarme è attivo sul telefonino, c'è bisogno anche di un programma Python da caricare sullo smartphone Android. Ma per poterlo fare bisogna innanzitutto installare una App che possa accedere al sottostante ambiente Linux (che in realtà è la base di Android),

come per esempio "Termux".

Tranquilli perché non richiede i privilegi di Root, ma apre semplicemente una emulazione terminale Linux. Utilizzando questo accesso possiamo installare Python sul telefono e poi la libreria Python per Telegram ("Telethon"). Inoltre è il caso di installare anche "play-audio" in Linux con il solito comando "pkg install play-audio". Play-audio serve per eseguire il file mp3 di allarme, che è compreso nei file del sistema, ma che potete cambiare con uno di vostra scelta.

Tutto il software commentato lo potete trovare su https://github.com/open-electronics/Artificial_Intelligence/tree/master/Deep_Learning/YoloPython/OpenCvYoloAlarm. Qui lo descriviamo a grandi linee. Il modulo operativo vero e proprio, cioè quello che carica la rete e la esegue corrisponde al file "OpencvYoloAlarm.py".

Questo programma verifica il caricamento della rete e l'accesso alla webcam, quindi ripete un ciclo infinito in cui:

- legge una frame (con opencv);
- la passa alla routine di pre-trattamento (scalatura 416x416 dell'immagine e trasformazione in monocromatica);
- la carica come input della rete ed esegue il "forwarding" fino ad avere il risultato di output;
- elabora il risultato per isolare gli oggetti ed inquadrarli sulla frame stampando anche il livello di confidenza;
- inserisce la frame in una coda perché possa essere letta dal web-server;
- controlla se è presente almeno un oggetto persona; se sì, attiva la procedura di allarme;
- se è scattato un allarme si predispose ad ignorare altri allarmi per 60 secondi.

Il file "Common.py" è un file che centralizza l'accesso ai parametri (letti dal file di configurazione) e crea la coda delle frame. Una piccolissima libreria ("GetNetAddressLib.py") definisce la funzione che determina gli indirizzi di rete locale e quello di rete pubblica del sistema in cui si trova ad essere invocata.

Questi indirizzi saranno spediti come messaggio Telegram in caso di allarme. In questo modo, se il telefono si trova a portata WiFi dal router Internet, potrà aprire il browser cliccandoci sopra. Se invece si trova al di fuori della rete locale può raggiungere il server web utilizzando l'indirizzo pubblico a patto che in fase di installazione del sistema ci si sia preoccupati di intervenire sul router assegnando la porta dell'applicazione (4000)

al forwarding del sistema jetson così come viene visto in ambito locale. Il fatto di spedire ogni volta l'indirizzo pubblico evita di doversi affidare ad un server DDNS.

Come noto, infatti, gli indirizzi pubblici possono cambiare in base alle esigenze dei provider, ed è il motivo per cui si provvede con i DDNS, che tengono traccia della modifica e gli assegnano comunque lo stesso nome pubblico.

In caso di allarme, il sistema si preoccupa anche di registrare un breve filmato (10 secondi) che viene comunque sovrascritto ad ogni allarme per evitare la saturazione di Jetson. Questi filmati vengono spediti come messaggio video e quindi rimangono sul server Telegram finché non vengono cancellati dall'utente. Inoltre il sistema registra un file di log (con gli eventi).

Per evitare la saturazione di Jetson, anche questo file è sovrascritto ma in modo circolare. Ovvero l'ultimo diventa il precedente mentre viene aperto un nuovo ultimo. Per cui al massimo sono presenti solo due file di 64KB.

Il modulo che realizza il server web si basa sulle librerie standard di Python (versione 3.6) e utilizza un risponditore (HTTPRequestHandler) personalizzato nella routine che gestisce il metodo "POST". Infatti la pagina html del sito è stata costruita con funzioni Javascript che dialogano con il server sulla base di precise richieste "POST". La pagina visualizza la webcam (con gli oggetti evidenziati) e può attivare o disattivare l'allarme telefonico.

CONCLUSIONI

Sul sito https://github.com/open-electronics/Artificial_Intelligence/tree/master/Deep_Learning/YoloPython oltre all'applicazione allarme, potete trovare altri due semplici programmi:

- OpencvYoloCamera.py è un programma Python che visualizza in tempo reale il riconoscimento di oggetti visti dalla telecamera; è possibile inserire diversi parametri dalla riga di comando. Inserendo -h si ottiene l'help;
- OpencvYoloPhoto.py è un programma Python che, invece, permette di selezionare un'immagine sul computer, nella quale riconosce i vari oggetti.

Tenete però presente che hanno bisogno della directory darknet (che contiene le strutture e i pesi delle versioni YOLO) che potete trovare nella forma di link automatico per il downloading da un altro server a causa delle dimensioni massime permesse da GitHub. 





IMPARA L'ELETTRONICA FACILMENTE CON TOTEM MINI LAB



€ 112,00
cod. TOTEMMINILAB

Piccolo ma potente laboratorio di elettronica ideale per sperimentare e apprendere le basi dell'elettronica. Pienamente compatibile con l'ambiente di programmazione Arduino, Totem Mini Lab comprende una varietà di strumenti di misura e blocchi elettronici pronti all'uso per realizzare esperimenti. Pensato per essere utilizzato insieme al sistema di costruzione Totem consente agli utenti di costruire solidi banchi da lavoro con breadboard integrate. Inoltre, il laboratorio è compatibile con schede di espansione I/O, sensori e audio che possono espandere ulteriormente le capacità del Mini Lab. L'utilizzo di Totem Mini Lab è semplice e divertente, ideale sia per ragazzi sia per adulti!

COMPLETALO CON LE SCHEDE DI ESPANSIONE!



ESPANSIONE AUDIO

Con questo pannello sarà possibile avere disponibili e facilmente accessibili i seguenti componenti: amplificatore audio in classe AB con 1 W di potenza, altoparlante integrato e alimentazione a 12 volt, VU-meter con barra a 10 LED, generatore di funzioni in grado di generare segnali fino a 1 MHz.

€ 39,50
Cod. TOTEMAUDIOSIDE



ESPANSIONE I/O

Contiene componenti elettronici che solitamente sono difficili da inserire sulle breadboard. Con questo pannello sarà possibile avere disponibili e facilmente accessibili i seguenti componenti: 3 interruttori a pulsante con indicatori LED, 3 potenziometri da 10 kOhm, 1 encoder rotativo, 3 pulsanti, 1 LED RGB, 1 presa Jack audio da 3,5 mm, 1 Relè da 10 A / 250 V, con LED di stato.

€ 37,00
Cod. TOTEMIOSIDE



ESPANSIONE SENSORI

Con questo pannello sarà possibile avere disponibili e facilmente accessibili i seguenti componenti: un microfono preamplificato, un display OLED 128x64 con interfaccia I²C, un sensore umidità DHT11, un termistore NTC da 10 kohm, un buzzer e un piccolo driver per motori DC.

€ 39,50
Cod. TOTEMSENSORSIDE



Utilizziamo il sensore MPU6050 incorporato nella nostra demoboard per misurare i dati dell'accelerometro e del giroscopio e visualizzarli in Azure IoT Central. Sesta e ultima puntata.

CONOSCERE

E USARE

Azure Sphere

6

di PIER CALDERAN

Ebbene, il corso che abbiamo dedicato alla tecnologia Azure Sphere volge al termine e si conclude in queste pagine; vi abbiamo introdotto la materia, descrivendovi la piattaforma Cloud che Microsoft ha realizzato e che rende disponibile per l'implementazione di applicazioni Big Data e Internet of Things, quindi abbiamo focalizzato la nostra attenzione su una piattaforma di sviluppo implementata dalla scheda di Futura Elettronica creata ad hoc e dotata dell'hardware che serve a comunicare tramite rete WiFi per trasferire i dati di numerosi sensori alla piattaforma stessa. Con quella piattaforma abbiamo sviluppato e proposto, puntata dopo puntata, varie applicazioni mirate a dimostrare cosa si possa fare disponendo di hardware edge IoT e appoggiandosi ad Azure Sphere mediante la creazione di account specifici, sia liberi che onerosi.

Vogliamo quindi concludere questa carrellata con l'applicazione spiegata in queste pagine, che ci permetterà di leggere i dati del sensore MPU6050 e di inviarli a IoT Central. Il codice di esempio di cui parliamo oggi si trova nella cartella Futura_samples\Futura_MT3620_MPU6050_IoT_Central. Per questo e tutti gli altri esempi di codice, dovete clonare il repository dal GitHub https://github.com/piercalderan/futura_MT3620.

IL SENSORE MPU6050

Il chip MPU6050 di InvenSense (invensense.tdk.com) è il primo dispositivo MotionTracking al mondo progettato per i requisiti di bassa potenza, basso costo e alte prestazioni per applicazioni di ogni tipo. L'MPU6050 sfrutta la tecnologia MEMS (Micro Electro-Mechanical Systems) e incorpora il firmware MotionFusion di InvenSense, che

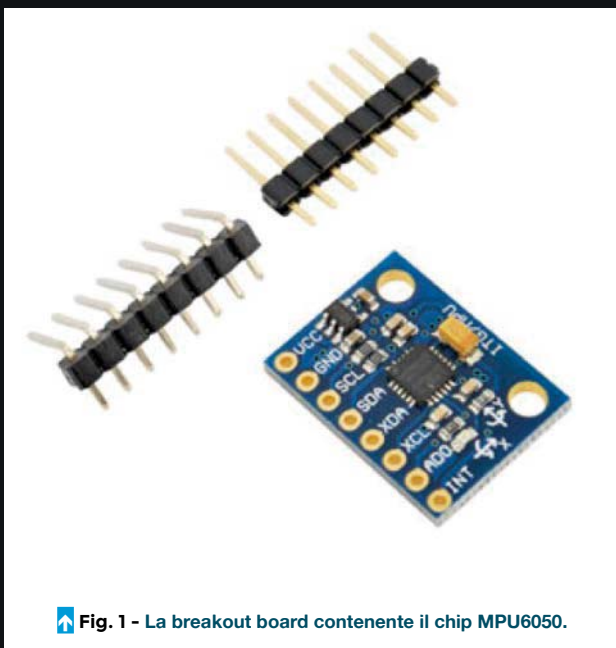


Fig. 1 - La breakout board contenente il chip MPU6050.

consente di eliminare l'integrazione con altri dispositivi discreti, garantendo che gli algoritmi di fusione dei sensori e le procedure di calibrazione forniscano prestazioni elevate per l'utente finale.

I dispositivi MPU6050 combinano un giroscopio a 3 assi e un accelerometro a 3 assi sullo stesso chip, insieme a un Digital Motion Processor (DMP) integrato, che elabora complessi algoritmi MotionFusion a 6 assi.

Il dispositivo può accedere a magnetometri esterni o altri sensori tramite un bus I2C master ausiliario, consentendo ai dispositivi di raccogliere una serie completa di dati del sensore senza intervento del processore del sistema.

Per il tracciamento di precisione di movimenti sia veloci che lenti, le parti sono dotate di un intervallo completo del giroscopio programmabile dall'utente di ± 250 , ± 500 , ± 1000 e ± 2000 °/sec (dps) e di un accelerometro programmabile dall'utente intervallo scala di $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$.

Altre caratteristiche includono un sensore di temperatura incorporato e un oscillatore su chip con variazione di $\pm 1\%$

sull'intervallo di temperatura di esercizio.

Le specifiche del chip e il protocollo di comunicazione I²C sono spiegati in dettaglio nel documento MPU6050.pdf che trovate all'interno della cartella del progetto (del corso Azure Sphere...) sul nostro sito www.elettronica.in.

Il chip MPU6050 viene venduto su breakout board (quella che vedete nella Fig. 1) e con due strip maschi a passo 2,54 mm, uno dritto e uno a 90°, da saldare.

La scheda Futura MT3620 può ospitare facilmente una breakout board MPU6050 collegata al bus I²C definito nella definizione hardware come SAMPLE_ISUO_I2C e corrispondente ai pin fisici 19 e 20, ovvero alle porte GPIO27 e GPIO28.

IL BUS I²C

Il nome esatto del bus è I²C e andrebbe pronunciato

"i-quadro-ci". È la sigla per "Inter Integrated Circuit".

Si tratta di un'interfaccia di comunicazione dati seriale a due linee utilizzata per la trasmissione/ricezione diretta di dati tra due o più circuiti integrati; non è una connessione fra due dispositivi ma un bus cui possono essere collegati tra loro in parallelo molti dispositivi.

La configurazione tipica dell'I²C-Bus è formata da un circuito master e uno o più circuiti slave, come illustrato nella Fig. 2.

Per il collegamento master-slave è necessaria la presenza di due resistori di pull-up che di solito non servono perché già previsti nei dispositivi, per cui sono sufficienti due soli fili per la connessione e una massa.

Il protocollo hardware dell'I²C si basa su due linee che sono SDA (Serial Data) e SCL (Serial Clock): la prima veicola i dati bidirezionalmente da e verso i dispositivi, mentre la seconda trasporta il clock.

I dati seriali sono sincronizzati al clock del dispositivo master, la cui velocità può essere di 100 kbit/s o di 400 kbit/s a seconda dell'impostazione della modalità Low Speed o Fast Speed. Gli indirizzi usati sono in totale 128, ma 16 di questi sono riservati, per cui al bus possono essere collegati al massimo 112 dispositivi in parallelo.

Le resistenze di pull-up hanno una loro influenza sulla massima velocità di comunicazione raggiungibile, perché formano con i dispositivi delle costanti di tempo che rallentano la

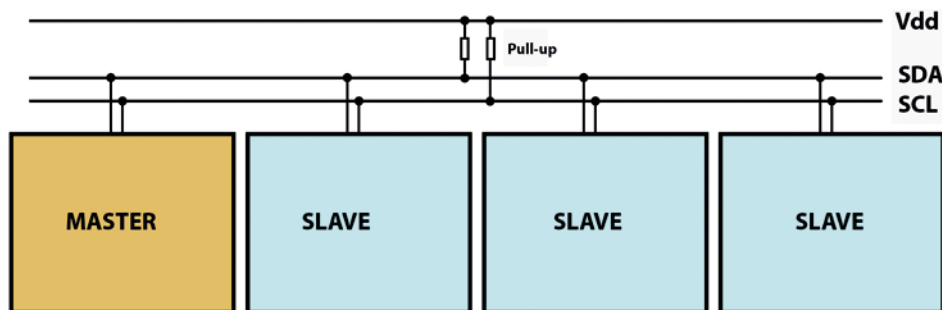


Fig. 2 - Il bus I2C.

salita del segnale da 0 logico a 5V; va infatti ricordato che i dispositivi, per non danneggiarsi l'uno con l'altro, hanno le uscite di tipo open-collector (oppure open-drain) in modo che quella che porta la linea a massa (zero logico) non carichi le altre, le quali restano in stato open.

LA LIBRERIA I²C

Ciò detto, possiamo ad analizzare la libreria che permette la gestione del bus. Il progetto descritto in queste pagine include la libreria di sistema "I2C" appositamente progettata per MT3620. Va inclusa all'inizio del codice con l'istruzione:

```
#include <applib/i2c.h>
```

La libreria gestisce solamente il bus I²C e i dispositivi ad esso collegati vanno indirizzati manualmente, senza causare conflitti fra di loro. Per esempio, l'indirizzo dell'MPU6050 è stato fissato dal produttore come 0x68 per cui va dichiarato all'inizio del codice come costante globale:

```
static const uint8_t MPU6050Address = 0x68;
```

Vista l'enorme quantità di dispositivi I²C, la parte più difficile è trovare librerie che si adattino a particolari hardware, come, nel nostro caso il chip MT3620. Quindi tutto il codice che trovate per il sensore MPU6050 è frutto dell'analisi del datasheet del produttore. Per comodità abbiamo inserito nella cartella del progetto i PDF del datasheet e della mappa dei registri a cui si fa riferimento in questo esempio.

I REGISTRI DELL'MPU6050

Il codice per l'MPU6050 è stato scritto appositamente tenendo conto dei registri dichiarati dal costruttore. Per altri dati fate riferimento al documento "MPU6050-Register-Map.pdf" che si trova nella cartella del progetto. Quello che vi serve sapere prima di tutto sono i comandi da inviare ai registri per configurare correttamente l'MPU6050. Qui sotto potete vedere alcuni registri cui fare riferimento:

```
#define MPU6050_ADDR      0x68 //MPU6050 address
#define MPU6050_SMPLRT_DIV 0x19 //Sample Rate Divider
#define MPU6050_CONFIG    0x1a //Configuration
#define MPU6050_GYRO_CONFIG 0x1b //Gyroscope Configuration
#define MPU6050_ACCEL_CONFIG 0x1c //Accelerometer Configuration
#define MPU6050_WHO_AM_I  0x75 //Who Am I
#define MPU6050_PWR_MGMT_1 0x6b //Power Management 1
#define MPU6050_TEMP_H    0x41 //Temperature Measurement H
#define MPU6050_TEMP_L    0x42 //Temperature Measurement L
```

A questo punto, dovete inviare i comandi ai suddetti registri. Per esempio, la riga per il comando 1 del registro "Sample Rate Divider" è la seguente:

```
const uint8_t Command1[] = { MPU6050_SMPLRT_DIV, 0x00 };
```

```
ssize_t transferredBytes1 = I2CMaster_Write(i2cFd, MPU6050Address, Command1, sizeof(Command1));
```

Per gli altri comandi la procedura è identica, cambia solo il registro. Sfogliando ancora la mappa dei registri, bisogna identificare i registri per ogni asse e il sensore di temperatura.

In questo modo si potrà interrogare il dispositivo e ricevere i byte High e Low che determinano la risposta. I registri per leggere la temperatura, i tre assi dell'accelerometro e i tre assi del giroscopio sono definiti in questo modo:

```
static const uint8_t outTmp = 0x41; //Temperature
static const uint8_t outAcX = 0x3b; //Accelerometer X
static const uint8_t outAcY = 0x3d; //Accelerometer Y
static const uint8_t outAcZ = 0x3f; //Accelerometer Z
static const uint8_t outGyX = 0x43; //Gyroscope X
static const uint8_t outGyY = 0x45; //Gyroscope Y
static const uint8_t outGyZ = 0x47; //Gyroscope Z
```

Un breve esempio per conoscere la temperatura del sensore MEMS a bordo dell'MPU6050.

Prima di tutto, dovete scrivere nel registro 0x41 (della variabile outTmp) per leggere i due byte High e Low di risposta:

```
uint8_t zRaw, zRaw1;
transferredBytes = I2CMaster_Write(i2cFd, MPU6050Address,
&outTmp, sizeof(outTmp));
ssize_t hi = I2CMaster_Read(i2cFd, MPU6050Address,
(uint8_t*)&zRaw, sizeof(zRaw));
ssize_t lo = I2CMaster_Read(i2cFd, MPU6050Address,
(uint8_t*)&zRaw1, sizeof(zRaw1));
```

Una volta ricevuti i due byte, si devono convertire in un intero e ottenere il valore di temperatura in forma "grezza" (raw):

```
Tmp = zRaw << 8 | zRaw1; //0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
```

Il fattore di scala e l'offset per il sensore di temperatura si trovano nella tabella delle specifiche elettriche (sezione 6.4 del documento delle specifiche del prodotto

 **Tabella 1 - Accelerazioni e valori dell'MPU6050.**

ACCELERAZIONE LINEARE	VALORE
FS = +/-2g	0.061
FS = +/-4g	0.122
FS = +/-8g	0.244
FS = +/-16g	0.732

↓ Listato 1

```
static void SendAccelButtonHandler(void)
{
    if (IsButtonPressed(sendMessageButtonGpioFdAccel, &sendMessageButtonStateAccel)) {
        char tempBuffer[20];
        int len = snprintf(tempBuffer, 20, "%d", AcX);
        if (len > 0) {
            SendTelemetry("Acce1X", tempBuffer);    //AcX
            Log_Debug("AcX: %s\n", tempBuffer);
        }

        len = snprintf(tempBuffer, 20, "%d", AcY);
        if (len > 0) SendTelemetry("Acce1Y", tempBuffer);    //AcY
        Log_Debug("AcY: %d\n", AcY);

        len = snprintf(tempBuffer, 20, "%d", AcZ);
        if (len > 0) SendTelemetry("Acce1Z", tempBuffer);    //AcZ
        Log_Debug("AcZ: %d\n", AcZ);

        len = snprintf(tempBuffer, 20, "%d", GyX);
        if (len > 0) SendTelemetry("GyroX", tempBuffer);    //GyX
        Log_Debug("GyX: %d\n", GyX);

        len = snprintf(tempBuffer, 20, "%d", GyY);
        if (len > 0) SendTelemetry("GyroY", tempBuffer);    //GyY
        Log_Debug("GyY: %d\n", GyY);

        len = snprintf(tempBuffer, 20, "%d", GyZ);
        if (len > 0) SendTelemetry("GyroZ", tempBuffer);    //GyZ
        Log_Debug("GyZ: %d\n", GyZ);

        len = snprintf(tempBuffer, 20, "%.21f", temp_MPU6050);
        if (len > 0) SendTelemetry("TempMPU6050", tempBuffer);    // tempMPU6050
        Log_Debug("Temp MPU6050: %i\n", tempBuffer);
    }
}
```

MPU6000/MPU6050) per cui il valore esatto della temperatura va calcolato con questa operazione:

```
// Temperature in degrees C = (TEMP_OUT Register Value as a
signed quantity)/340 + 36.53
double temp = Tmp / 340.00 + 36.53;
```

Questo significa che dovete dividere il valore della temperatura per 340 e aggiungere un offset di 36,53.

Nella finestra di Output potrete vedere la temperatura in gradi con due cifre decimali:

```
Log_Debug("Temp: %.21f\n", temp);
```

FUNZIONE SendAccelButtonHandler

Con la pressione del pulsante 2 viene chiamata la funzione "SendAccelButtonHandler", che a sua volta chiama la funzione "SendTelemetry" per ognuno degli assi e della tempera-



↑ Fig. 3 - La pagina "Selezionare il tipo".



↑ Fig. 4 - La pagina "Importa un modello".

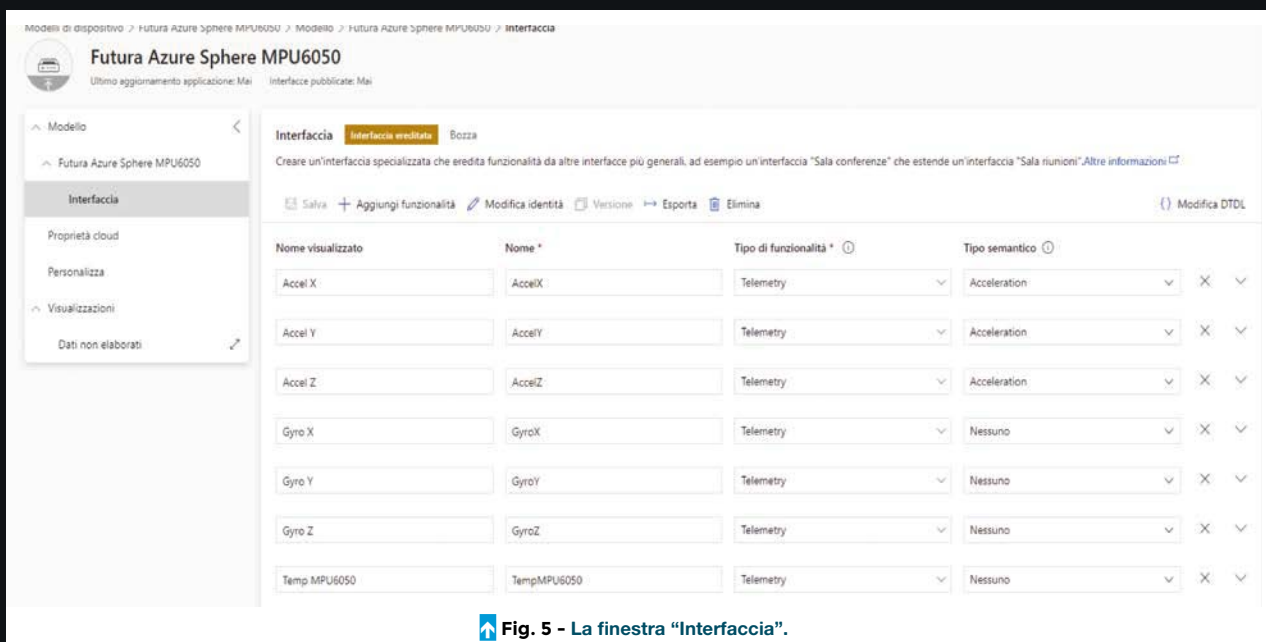


Fig. 5 - La finestra "Interfaccia".

tura dell'MPU6050.

Notare che i nomi dei campi sono stati abbreviati in questo modo:

AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

Ovvero:

Accelerometro asse x

Accelerometro asse y

Accelerometro asse z

Temperatura

Giroscopio asse x

Giroscopio asse y

Giroscopio asse z

Ovviamente, quando andrete a costruire il dispositivo su IoT Central dovreste ricordare di usare gli stessi nomi dei campi nella dashboard del modello del dispositivo.

Si fa notare che i dati di accelerazione vengono mandati "grezzi" così come vengono letti.

Se si vogliono convertire i dati in g (gravità), si può fare riferimento all'operazione seguente, tenendo conto di $FS = +/-4g$ (vedere la **Tabella 1** per il valore da utilizzare in base all'accelerazione):

$$\text{double } g = (zRaw * 0.122) / 1000.0;$$



Fig. 6 - La pagina "Selezionare per aggiungere una nuova visualizzazione".

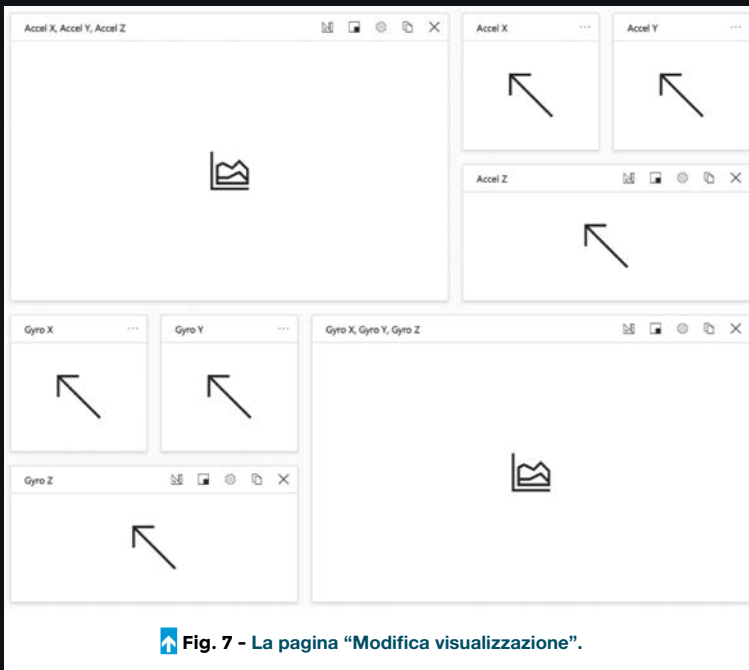


Fig. 7 - La pagina “Modifica visualizzazione”.

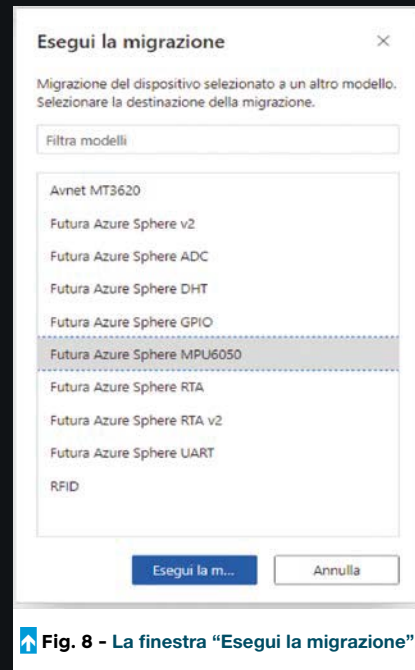


Fig. 8 - La finestra “Esegui la migrazione”.

I dati vengono ricevuti dall'MPU6050 con un timer impostato a 0,9 secondi:

```
static const struct timespec accelReadPeriod =
{ .tv_sec = 0, .tv_nsec = 900000000 };
```

Ovviamente potete modificare il timer di polling a piacere. Per evitare un intasamento nel server dovuto a un eccesso di dati e anche il conseguente consumo di risorse disponibili, la funzione di invio, avviene solo dopo aver premuto il pulsante 2 (**Listato 1**).

IL FILE APP_MANIFEST

L'invio dei dati dell'MPU6050 avviene premendo il pulsante 2, per cui dovete impostare nelle “Capabilities” del file app_manifest.json le definizioni hardware “SAMPLE_BUTTON_2” per il pulsante 2 e “SAMPLE_ISU0_I2C” per il bus I2C master:

Gli argomenti “CmdArgs”, “AllowedConnections” e “DeviceAuthentication” sono gli stessi degli esempi già pubblicati nelle scorse puntate del corso:

```
{
  "SchemaVersion": 1,
  "Name": "AzureIoT",
  "ComponentId": "819255ff-8640-41fd-aea7-f85d34c491d5",
  "EntryPoint": "/bin/app",
  "CmdArgs": [ "0ne001C85E4" ],
  "Capabilities": {
    "AllowedConnections": [
      "global.azure-devices-provisioning.net",

```

```
      "iotc-f18c6479-0fd2-4197-80f0-337ba9198b77.azure-devices.net"
    ],
    "Gpio": [ "$SAMPLE_BUTTON_2" ],
    "I2cMaster": [ "$SAMPLE_ISU0_I2C" ],
    "DeviceAuthentication": "df7ba4db-4a32-4bfd-b401-59f2846cc320"
  },
  "ApplicationType": "Default"
}
```

LA TELEMETRIA DELL'MPU6050 SU IoT CENTRAL

Questa volta, invece delle procedure di creazione da zero del modello di dispositivo, potete trovare nella directory dell'esempio il modello già pronto all'uso; questo vi risparmierà la fatica di dover inserire tutte le funzionalità dei 6 assi e della temperatura dell'MPU6050.

Ecco come dovete procedere:

1. selezionate “Modelli di dispositivo”;
2. create un nuovo modello con “+ Nuovo” allorché apparirà la finestra “Selezionate il tipo”;
3. selezionate “Dispositivo “IoT” (**Fig. 3**);
4. fate clic su “Successivo: Personalizza” allorché apparirà la finestra “Personalizza”;
5. compilate il campo Nome visualizzato: “Futura Azure Sphere MPU6050”;
6. fate clic su “Avanti: Rivedi” e vi apparirà la finestra “Visualizza”;
7. fate clic su “Crea” e così facendo verrà creato il modello “Futura Azure Sphere MPU6050”;
8. fate clic su “Importa un modello” (**Fig. 4**);

9. dalla finestra di dialogo cui accederete, cercate nella cartella Futura_MT3620\Futura_samples il file: "Futura Azure Sphere MPU6050.json" e caricatelo;
10. nella la pagina "Interfaccia" che si aprirà, fate clic su "Interfaccia" del pannello a sinistra.

Al termine della procedura si apriranno tutte le funzionalità di telemetria dell'MPU6050 importate dal file e pronte all'uso (Fig. 5):

Accel X
Accel Y
Accel Z
Gyro X
Gyro Y
Gyro Z
Temp MPU6050

Ora, aggiungete la visualizzazione dei dati MPU6050 andando su "Visualizzazioni"; facendo ciò, si aprirà la pagina "Selezionate per aggiungere una nuova visualizzazione" (Fig. 6). Da qui, selezionate "Genera visualizzazioni predefinite".

Dalla pagina "Selezionate le visualizzazioni applicabili da generare" fate clic su "Genera visualizzazioni dashboard predefinite".

In questo modo, verranno create le visualizzazioni standard per tutte le funzionalità.

Fate clic su "Overview" sotto "Visualizzazioni" e si aprirà finalmente la pagina con tutti i riquadri preconfigurati (Fig. 7). Qui potete vedere il riquadro con il gruppo "Accel X", "Accel Y" e "Accel Z" e il riquadro con il gruppo "Gyro X", "Gyro Y" e "Gyro Z" e i riquadri KPI di tutti gli assi separati.

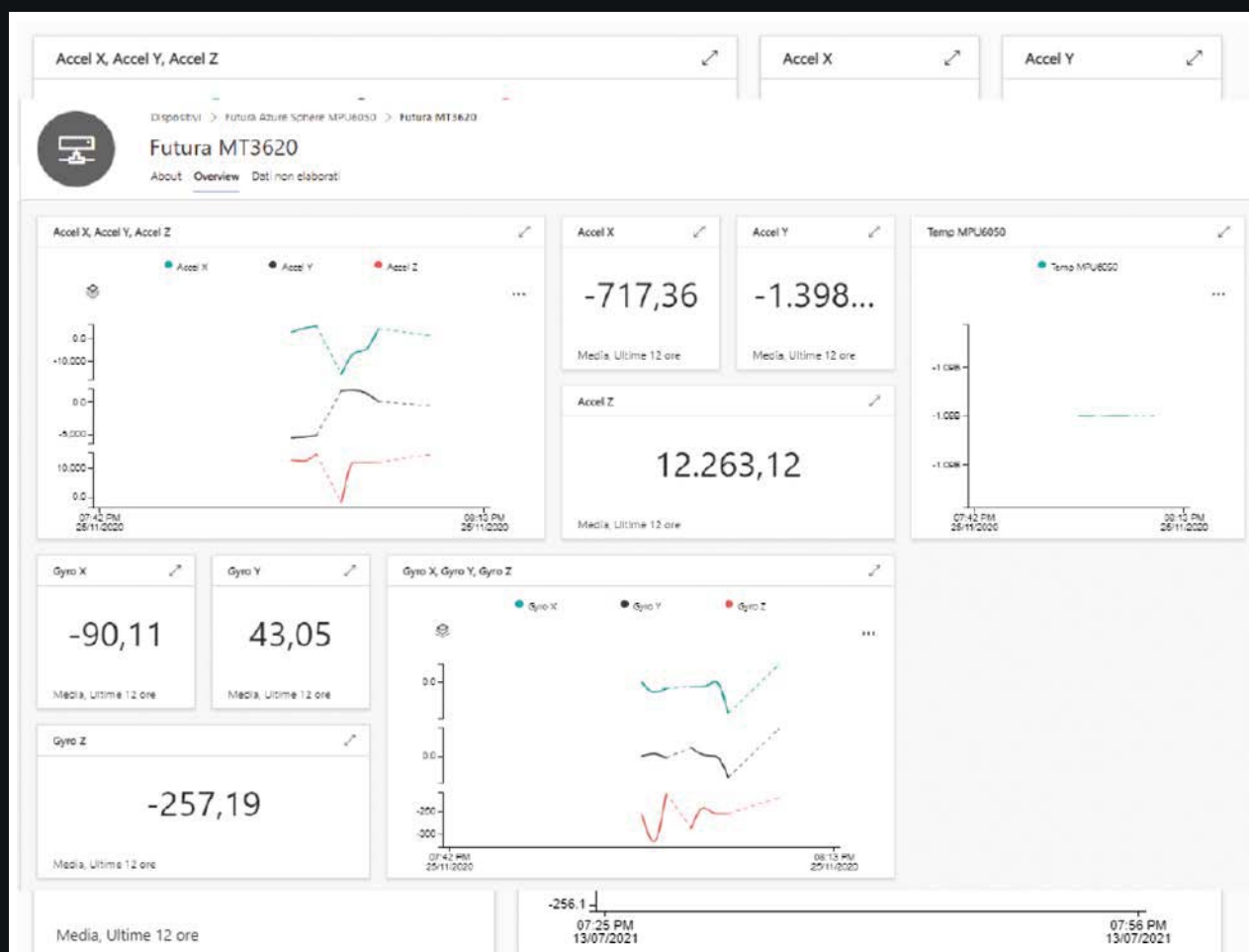
Manca solo il riquadro della temperatura, perché non previsto dall'automazione del wizard.

Pertanto, fate clic su "Seleziona i dati di telemetria", selezionate "Temp MPU6050" e premere il tasto "Aggiungi riquadro".

Dato che le visualizzazioni sono sempre in bozza, potete modificare i riquadri a piacere.

Al termine delle modifiche, ricordatevi di salvare con il pulsante "Salva" e pubblicate il modello, come al solito.

Nella pagina "Dispositivi", selezionate "Futura MT3620" ed effettuate la migrazione al nuovo modello "Futura Azure Sphere MPU6050" come indicato nella Fig. 8.



➔ Fig. 9 - La pagina del dispositivo di IoT Central.

AVVIO DELL'APPLICAZIONE

Prima di avviare l'applicazione da Visual Studio, se volete evitare di vedere una quantità enorme di dati in arrivo dall'MPU6050 ogni secondo, mettete in commento le righe qui sotto, che si trovano all'interno della funzione *static void AccelerometerHandler(EventLoopTimer* timer)*:

```
//Log_Debug("Temp: %.2lf\n", temp_MPU6050);
//Log_Debug("AcX: %d\n", AcX);
//Log_Debug("AcY: %d\n", AcY);
//Log_Debug("AcZ: %d\n", AcZ);
//Log_Debug("GyX: %d\n", GyX);
//Log_Debug("GyY: %d\n", GyY);
//Log_Debug("GyZ: %d\n", GyZ);
```

All'avvio dell'applicazione, vedrete nella finestra di Output qualcosa di simile a questo:

```
I2C MPU6050 application starting.
Setting Azure Scope ID 0ne001C85E4
Opening SAMPLE_BUTTON_2 as input
Open ok
Set Speed ok
Set Time Out ok
Set Default Target Address ok
[Azure IoT] Using HSM cert at /run/daa/df7ba4db-4a32-4bfd-b401-59f2846cc320
IoTHubDeviceClient_LL_CreateWithAzureSphereDeviceAuthProvisioning
returned 'AZURE_SPHERE_PROV_RESULT_OK'.
IoT Hub Authenticated: IOTHUB_CLIENT_CONNECTION_OK
Sending IoT Hub Message: { "AcceIX": "509" }
INFO: IoTHubClient accepted the message for delivery
```

Solo quando premete il pulsante 2, vedrete i dati inviati a IoT Central e la risposta del server:

```
Sending IoT Hub Message: { "AcceLY": "-716" }
INFO: IoTHubClient accepted the message for delivery
AcY: -716
Sending IoT Hub Message: { "AcceLZ": "13558" }
INFO: IoTHubClient accepted the message for delivery
AcZ: 13558
Sending IoT Hub Message: { "GyroX": "-1" }
INFO: IoTHubClient accepted the message for delivery
GyX: -1
Sending IoT Hub Message: { "GyroY": "-1" }
INFO: IoTHubClient accepted the message for delivery
GyY: -1
Sending IoT Hub Message: { "GyroZ": "-256" }
INFO: IoTHubClient accepted the message for delivery
GyZ: -256
Sending IoT Hub Message: { "TempMPU6050": "29.75" }
INFO: IoTHubClient accepted the message for delivery
Temp MPU6050: -1090528272
```

```
INFO: Message received by IoT Hub. Result is: 0
INFO: Message received by IoT Hub. Result is: 0
INFO: Message received by IoT Hub. Result is: 0
INFO: Message received by IoT Hub. Result is: 0
INFO: Message received by IoT Hub. Result is: 0
INFO: Message received by IoT Hub. Result is: 0
INFO: Message received by IoT Hub. Result is: 0
```

Quello che vedrete nel dispositivo di IoT Central è illustrato nella schermata proposta dalla Fig. 9.

CONCLUSIONI

Bene, come accennato a inizio di questa puntata, si conclude qui questo corso dedicato all'utilizzo e all'integrazione di Microsoft Azure Sphere. A beneficio di chi avesse iniziato a seguire la materia solo da questo numero riepiloghiamo brevemente ciò che è stato esposto nelle precedenti cinque puntate: nella prima abbiamo introdotto la piattaforma di sviluppo hardware Futura Elettronica (Demo Board Azure Sphere MT3620) e creato una sorta di hello world, nella seconda abbiamo introdotto la piattaforma Cloud IoT Central su cui raccogliere i dati, spiegando come e a che condizioni creare un account utente, ma anche come registrare e configurare un dispositivo che dobbiamo interfacciare ad essa tramite Internet e la connessione WiFi. Nella terza puntata abbiamo spiegato come fare il provisioning delle applicazioni da interfacciare con IoT Central.

Nella quarta puntata abbiamo trasmesso i dati ricevuti dalla UART su IoT Central, mentre nella quinta abbiamo proposto un'applicazione che permette di leggere i tag RFID tramite il lettore presente nella scheda di sviluppo e di inviarne i dati ad IoT Central. In questo corso abbiamo inoltre testato applicazioni demo con i sensori della scheda Azure Sphere MT3620 e terminiamo qui con la telemetria relativa a quelli affacciati all'I²C-Bus. □



Cosa occorre?

La demoboard di Futura Elettronica con MT3620 (cod. FEAZURESPH) è disponibile a Euro 128,00 e include i componenti base per poter sperimentare con la tecnologia Azure Sphere. Dal sito futurashop.it è possibile acquistare anche le breakout board correlate alla demoboard. La board USB/TTL in grado di creare quattro porte seriali virtuali (cod. FT1497K) è in vendita a Euro 35,00, mentre la board adattatrice necessaria per programmare la demoboard (cod. FT1498K) è disponibile a Euro 25,00. 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>

Tutto sotto controllo!

Vuoi stare tranquillo anche quando sei fuori casa? Niente di più facile con le telecamere WiFi motorizzate!

Istruzioni in italiano!



TELECAMERA MOTORIZZATA IP WiFi - 2MPX - ONVIF

Telecamera IP a colori motorizzata con sensore d'immagine CMOS da 2MPX. È dotata di ottica fissa da 3,6 mm e di 11 LED IR che permettono di riprendere immagini perfette anche in condizioni di scarsa luminosità. Supporta la visione da remoto tramite smartphone con sistema Android e iOS. I video possono essere registrati sulla memoria dello smartphone tramite apposita APP, oppure su micro SD Card (max. 128GB - acquistabile separatamente). Dispone di microfono e di altoparlante che permettono di parlare e di essere ascoltati da remoto. Formato di compressione video H.264/MJPEG. Push di allarme su motion detection, possibilità di visualizzare dall'APP fino a 9 telecamere contemporaneamente.



€ 42,00
cod. CP728N

Prezzi IVA inclusa.

TELECAMERA Wi-Fi MOTORIZZATA CON LED DUAL LIGHT - AUDIO BIDIREZIONALE / SD CARD / ETHERNET

Telecamera motorizzata Wi-Fi a colori basata sulla tecnologia ONVIF. Può essere gestita da remoto, tramite APP gratuita, disponibile sia per sistemi Android che iOS. Grazie al sistema PTZ (PAN/TILT/ZOOM) di cui la telecamera è dotata, è possibile orientare, da remoto, l'inquadratura delle immagini in funzione delle proprie esigenze di ripresa. Funzione motion detection con notifica di push sul display dello smartphone. È dotata di 8 LED, dei quali 4 IR e 4 a luce bianca. Integra un microfono e uno speaker, che consentono alla telecamera di parlare e di essere ascoltati da remoto. Offre la possibilità di registrare le immagini riprese, sia sulla memoria interna dello smartphone sul quale è installata l'APP di gestione, sia sulla micro SD card inserita nell'apposito slot della telecamera. Viene fornita completa di staffa di fissaggio.



€ 79,00
cod. FR769



€ 49,00
cod. CP729

TELECAMERA IP WiFi MOTORIZZATA CON FUNZIONE SMART TRACKING MOTION 2MPX ONVIF

Mini telecamera IP a colori motorizzata dotata di ottica fissa da 3,6 mm, 6 LED IR, con funzione Smart Tracking Motion (monitoraggio del movimento). Il modulo Wi-Fi integrato consente di collegare la telecamera in remoto permettendo di visualizzare su PC e Smartphone (iOS, Android), quanto ripreso dal suo obiettivo che può ruotare sia orizzontalmente sia verticalmente. Consente inoltre di registrare video su PC oppure sulla micro SD card integrata (max 128 GB, acquistabile separatamente) e visualizzare dall'APP fino a 9 telecamere contemporaneamente. Grazie al microfono e all'altoparlante integrato è possibile comunicare con chi è vicino alla telecamera.





Shell avvia il più grande elettrolizzatore europeo di idrogeno

10 MW: questa è la portata dell'elettrolizzatore da poco avviato presso lo Shell Energy and Chemicals Park Rheinland, che produce idrogeno verde. Integrato nel consorzio europeo Refhyne e con il finanziamento della Commissione europea attraverso la Fuel Cells and Hydrogen Joint Undertaking (FCH JU), l'impianto è il primo ad utilizzare la tecnologia su così larga scala in una raffineria.

Sono in corso piani per espandere la capacità dell'elettrolizzatore da 10 a 100 MW nel sito vicino a Colonia, dove Shell intende anche produrre carburante per aviazione sostenibile (SAF) utilizzando energia rinnovabile e biomassa. Si tratta di un modello di produzione di energia a basse emissioni di carbonio che può essere replicato in tutto il mondo per alimentare l'autotrazione e l'industria.

L'elettrolizzatore utilizzerà elettricità rinnovabile per produrre fino a 1.300 tonnellate di idrogeno all'anno; inizialmente sarà utilizzato per produrre combustibili con una minore densità di carbonio e aiutare a decarbonizzare altre industrie. È stato prodotto da ITM power a Sheffield, nel Regno Unito, e comprende parti realizzate in Italia, Svezia, Spagna e Germania.

Una gran quantità di idrogeno è oggi richiesta in Germania dai processi industriali e viene prodotto dall'industria petrolchimica o da processi tradizionali; ecco perché l'elettrolizzatore ha la sua importanza, in quanto va a soddisfare l'esigenza di produrre idrogeno a emissioni zero. Progetti come Refhyne dimostrano come l'innovazione possa portare benefici sia all'ambiente che all'economia. L'idrogeno svolgerà un ruolo chiave nella mobilità elettrica e nell'industria, poiché ha un grande potenziale per la decarbonizzazione dei settori in cui non è possibile l'utilizzo diretto delle energie rinnovabili.

refhyne.eu

L'auto di Klein Vision: atterra in aeroporto e prosegue sulla strada

Mercoledì 30 giugno la "Aircar" dell'azienda slovacca Klein Vision ha effettuato il suo primo volo ufficiale tra due città, una volta atterrata in aeroporto ha ritratto le sue ali e sulle quattro ruote ha proseguito il proprio viaggio fino al centro della città. L'auto, lo avrete capito, è una vettura volante dotata di ali e superfici di controllo per superare distanze e territori poco praticabili e il suo volo di test è iniziato dall'aeroporto di Nitra, con atterraggio a Bratislava circa 35 minuti dopo. Una volta a terra, espletate le procedure di "conversione stradale" è uscita sulle ruote dal piazzale, sempre guidata da Stefan Klein, fondatore della Klein Vision, che dalla fine degli anni '80 progetta prototipi su prototipi di automezzi su ruote in grado anche di volare. Questo gioiello di tecnologia è poi andato nel centro di Bratislava dove una piccola folla di ammiratori e giornalisti l'attendeva per i festeggiamenti.

Aircar ha le sembianze di una compatta auto sportiva, convertibile in un piccolo aeroplano le cui quattro ruote fungono anche da carrello (non re-

traibile) per le manovre di atterraggio e decollo. La conversione da automobile ad aereo si attiva mediante un pulsante in plancia, che permette di estrarre (o ritrarre) le ali retraibili e un alettone posteriore mobile che funge da piano di coda. Prima del volo da Nitra a Bratislava, AirCar ha superato 40 ore di volo di prova, superando rigorosi quanto indispensabili test di manovrabilità. La propulsione stradale dell'Aircar è assicurata da un motore BMW da 160 cavalli alimentato a benzina, che tramite una presa di forza e un'apposita trasmissione può essere sconnesso dalle ruote e utilizzato per far girare l'elica propulsiva, situata posteriormente alla cabina.

Nel volo di test, l'auto volante ha raggiunto quota 2.500 metri e una velocità di crociera di 190 chilometri all'ora. Queste prestazioni sono consentite con un peso massimo trasportato di 200 chilogrammi, passeggeri e bagagli compresi, però la Klein Vision sta lavorando a una nuova versione della Aircar, equipaggiata con un motore più potente (300 CV) che garantirà una ve-



Vele gonfiabili per le navi cargo

Michelin ha creato speciali vele funzionanti come ali, con la struttura gonfiabile centinata che può essere elevata e direzionata per cogliere meglio l'energia del vento. Ora vuole adattare la tecnologia alle navi da carico, per sfruttare l'energia eolica per ridurre l'uso di carburante diesel e quindi le emissioni di gas serra. Il progetto si chiama Wing Sail Mobility, o WISAMO e il sistema si attiva con un pulsante: innanzitutto l'albero telescopico si alza dalla sua base, raggiungendo fino a 17 metri di altezza. L'ala gonfiabile in tessuto, della superficie di 93 metri quadri, si dispiega lentamente mentre un piccolo compressore la gonfia. Il vento che scorre sulle due superfici opposte determina, come in un'ala d'aereo, variazioni della pressione che creano portanza, aiutando a spingere la nave in avanti. Quando la nave si avvicina a un ponte o incontra condizioni meteorologiche avverse, il sistema si ritrae automaticamente. Michelin stima che l'ala possa ridurre il consumo del carburante di una nave fino al 20%, sulla base delle misurazioni di test tecnici e simulazioni; per una grande nave portacontainer, ciò potrebbe significare evitare di bruciare decine di migliaia di litri di carburante al giorno. Michelin prevede di testare la tecnologia su un mercantile commerciale nel 2022. WISAMO è un passo significativo verso l'ambizione di Michelin di decarbonizzare completamente la sua catena di approvvigionamento. Il progetto si unisce a una serie di iniziative di propulsione assistita dal vento in tutto il mondo. Sulla nave portarinfuse MV Afros, quattro rotori Flettner (o vele rotori-



che) ruotano a un ritmo vertiginoso, creando una spinta che muove la nave in avanti. La nave mercantile Lysbris Seaways è dotata di strutture non rotanti dotate di prese d'aria e ventilatori interni che aspirano il sottile strato d'aria che giace sulla superficie dell'ala. Il trasporto merci è responsabile di quasi il 3% delle emissioni mondiali annuali di gas serra e le navi sono una fonte significativa di inquinamento atmosferico tossico nei porti e nelle comunità costiere. Per le compagnie di navigazione, queste vele sono un modo per ridurre le emissioni senza stravolgere le navi esistenti e per limitare l'uso di carburante e ridurre i costi operativi (limitando anche le emissioni di CO₂). WISAMO è l'unica tecnologia a utilizzare ali gonfiabili.

www.michelin.com

locità di crociera di 300 km/h e un'autonomia di 1.000 chilometri, trasportando un peso maggiore.

Chiaramente un'auto volante non potrà essere alla portata di tutti gli automobilisti e non tanto per il costo eccessivo, quanto per il fatto che per volare non basterà la patente B: servirà anche essere in possesso di un brevetto da pilota per il

volo a vista. Al momento la Aircar rimane un concept e un veicolo dimostrativo, ma Klein Vision conta di commercializzarla nei prossimi anni, portandola sul mercato come si farebbe con una qualsiasi automobile; per farlo, tuttavia, bisogna sicuramente superare una serie di ostacoli, compresa l'industrializzazione di un prodotto simile e l'omologazione,

tuttavia la combinazione di tecnologie produttive e di soluzioni tecnologiche avanzate messe in campo negli ultimi anni può far diventare realtà il sogno di chi vuole avere un'auto volante che possa ammirare nel box di casa invece che al cinema, in un film di fantascienza.

www.klein-vision.com



Energia solare nel sale da cucina

L'azienda svizzera Innovenergy ha sviluppato una tecnologia di batteria alternativa pronta per la produzione in serie, che si basa esclusivamente su materie prime disponibili in grandi quantità e che potrebbe diventare un concorrente della batteria agli ioni di litio.

La AEW Energie AG dell'Argovia, una utility che in Svizzera fornisce elettricità alla regione e offre impianti fotovoltaici con accumulo di energia elettrica, ha implementato il primo sistema la cui batteria funziona con la tecnologia Innovenergy: a base di sale da cucina. È a prova di esplosione e non può bruciare. La densità di energia è paragonabile a quella della batteria agli ioni di litio. Funziona in modo affidabile in condizioni di freddo estremo e con



temperature esterne elevate. E difficilmente perde capacità anche dopo numerosi cicli di carica e scarica: dopo 15-20 anni avrà una durata molto più lunga rispetto alla Li-ion. Inoltre, non richiede alcuna manutenzione e alla fine può essere completamente riciclata.

L'unico svantaggio è che costa oltre 1/3 in più della batteria agli ioni di litio, che è già prodotta in serie. Tuttavia è un dispositivo più ecologico.

Le batterie al sale sono costituite dal 32% di normale sale da cucina. Contengono anche ferro, nichel e ceramica. Funzionano da -20°C a $+60^{\circ}\text{C}$, hanno all'incirca la stessa densità di energia di una batteria agli ioni di litio.

www.aew.ch



Il biosensore stampabile per tessuti e organi

I chirurghi potrebbero presto essere in grado di localizzare regioni critiche in tessuti e organi durante un'operazione, grazie a un nuovo biosensore della Purdue University stampato in 3D utilizzando un sistema di stampa automatizzato. Il biosensore, che consente la registrazione e l'imaging simultaneo di tessuti e organi durante l'intervento. La registrazione e l'imaging simultanei potrebbero essere utili durante la chirurgia cardiaca per guidare gli interventi chirurgici come una procedura per ripristinare i normali ritmi cardiaci.

I metodi tradizionali per la registrazione e l'immagine simultanea di tessuti e organi si sono rivelati difficili perché altri sensori utilizzati per la registrazione in genere interrompono il processo di imaging. Invece il nuovo biosensore è morbido, sottile ed estensibile, in grado di aderire perfettamente alla superficie curvilinea degli organi; ad esempio il cuore, anche durante le fasi di contrazione e rilascio.

Questa caratteristica unica consente, ad esempio, osservazioni in tempo reale sulla propagazione dell'infarto del miocardio in 3D.

Utilizzando bio-inchiostri morbidi durante la prototipazione rapida di un organo, i biosensori si adattano a una varietà di dimensioni e forme di un organo. I bio-inchiostri sono più morbidi del tessuto, si estendono senza subire la degradazione del sensore e hanno un'adesione naturale affidabile alla superficie bagnata degli organi senza bisogno di adesivi aggiuntivi.

www.purdue.edu

Lightyear One, l'auto elettrica fotovoltaica con 710 km di autonomia

Sulla pista di Aldenhoven, l'auto elettrica fotovoltaica Lightyear One con una sola ricarica ha percorso ben 710 km, valutati sul ciclo WLTP (Worldwide Harmonized Light Vehicles) usando una batteria di appena 60 kWh. I 710 km di autonomia sono stati ottenuti con un consumo energetico di soli 83 Wh/km alla velocità di 85 kmh. Anche le auto elettriche più efficienti oggi sul mercato consumano circa il 50% in più di energia a questa velocità.

Il veicolo può contare su un'autonomia tanto elevata sia grazie all'attenta progettazione e all'aerodinamica spinta,

sia perché incorpora su tettuccio e cofano 5 mq di pannelli fotovoltaici che aggiungono circa 70 km di autonomia, contribuendo alla ricarica della batteria di bordo insieme al sistema di frenata rigenerativa.

Dopo questo successo, inizierà la produzione di serie della vettura: i primi 946 veicoli dovrebbero entrare in produzione nella prima metà del 2022, per poi giungere sul mercato di massa a partire dal 2024, dove già suscitano una certa attesa.

lightyear.one



Scopri il mondo di ARDUINO e le sue infinite potenzialità!

Progetta, costruisci e divertiti con i moduli per Arduino predisposti per essere inseriti sui mattoncini della LEGO®. Puoi utilizzarli per robotica, con Arduino o Raspberry Pi, ecc. Nella confezione sono comprese, oltre ai sensori e agli accessori, anche una board Arduino Uno compatibile e una scheda Shield sperimentale dove puoi collegare in tutta semplicità i moduli sensori e attuatori. Una dispensa a corredo vi accompagna con gli sketch per testare i vari moduli con Arduino e la Sensor Shield.

*29 MODULI PER ARDUINO
COMPATIBILI CON LEGO®*

€ 79,00

Cod. ARDULEGOKIT

I MODULI PRESENTI NELLA CONFEZIONE:

- Modulo con display 7 segmenti a 4 cifre
- Modulo con sensore di fiamma
- Modulo con LED RGB 10 mm
- Modulo sensore di temperatura e umidità
- Modulo touch sensor
- Scheda sensore umidità terreno
- Scheda sensore livello liquidi
- Modulo con LED rosso
- Modulo con LED verde
- Modulo con LED giallo
- Modulo con sensore effetto di Hall
- Modulo con LED RGB SMD
- Modulo con sensore luce ambiente
- Modulo con pulsante rosso
- Modulo con pulsante giallo
- Modulo con interruttore a pulsante
- Modulo con buzzer senza elettronica
- Modulo sensore di vibrazione
- Modulo con sensore di temperatura
- Modulo con potenziometro
- Modulo con fotoresistenza
- Modulo con interruttore reed
- Modulo con ricevitore IR a 3 pin
- Modulo con buzzer con elettronica
- Modulo con encoder rotativo
- Modulo sensore scala di grigi
- Modulo con microswitch

Prezzi IVA inclusa



Starter kit V6 con Arduino Uno REV3



**IL PIÙ
COMPLETO!**

€ 65,00

Cod. ARDUKITV6

Acquista anche il kit
completo con tutti i
componenti necessari per
realizzare gli esperimenti
descritti nel libro "l'ABC
di ARDUINO".

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

