

CONOSCERE E USARE

di FRANCESCO FICILI

8

# Node-RED

**Impariamo ad usare Node-RED, un tool di flow-based programming orientato all'IoT ed alla connettività, originariamente sviluppato dall'IBM Emerging Technology Services team e adesso parte della JS Foundation. In questa puntata parliamo dell'integrazione degli assistenti vocali con Node-RED. Ultima Puntata.**

**C**ome abbiamo già visto ampiamente nella puntata precedente, Node-RED è facilmente interfacciabile con strumenti di utilizzo quotidiano, come ad esempio, per citarne solo alcuni, email, strumenti social ed applicazioni di messaggistica istantanea. Un altro oggetto con cui possiamo interfacciare agevolmente Node-RED sono gli assistenti vocali, un segmento di mercato di oggetti IoT che ha avuto una rapidissima crescita negli ultimi anni. In Italia parliamo di un valore di circa mezzo miliardo di euro, destinato ad aumentare vertiginosamente dal momento che gli assistenti vocali sono un enabler per un settore più ampio che possiamo genericamente definire come Smart Home, che vede la connessione di vari sensori ed attuatori all'interno delle nostre case per abilitare una gestione sempre più Smart dei dispositivi casalinghi. Esistono ormai diversi assistenti vocali disponibili sul mercato, sostanzialmente molto simili tra di

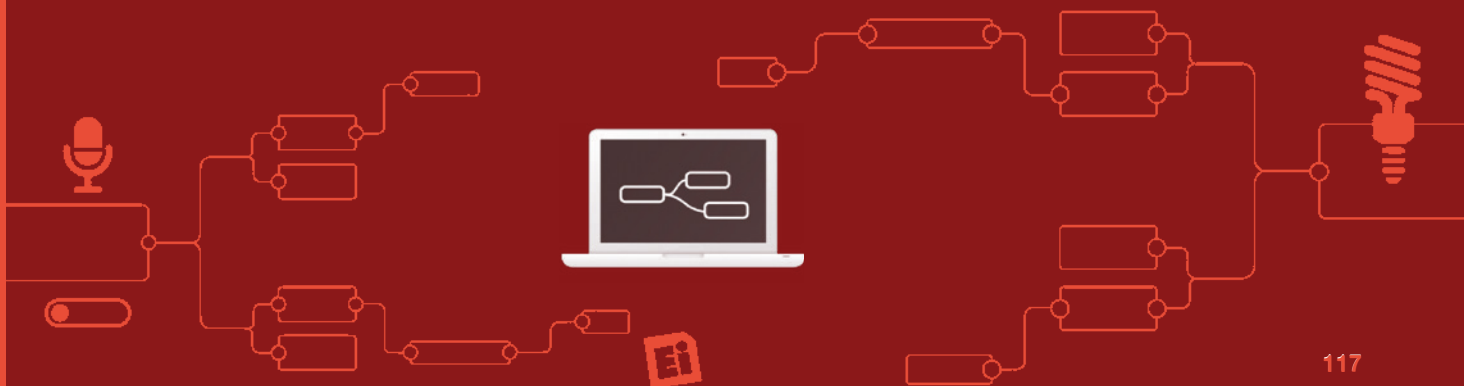
loro, sebbene ognuno con le sua peculiarità; i più comuni sono sicuramente gli assistenti vocali dei colossi Amazon e Google, ossia Alexa e Google Home. In questa puntata noi ci soffermeremo principalmente sui dispositivi della famiglia Alexa, tralasciando il caso di Google Home, che è comunque supportato da Node-RED.

## INTEGRAZIONE DI ALEXA IN NODE-RED

Esistono diversi nodi che possono essere utilizzati per integrare Alexa in Node-RED, e tutti questi nodi possono essere divisi sostanzialmente in due categorie:

- Nodi che permettono di emulare dispositivi controllabili via Alexa in Node-RED
- Nodi che permettono di interagire con le API di Alexa

I primi consentono di sfruttare le feature di controllo vocale di Alexa per controllare dei flussi. Si tratta essenzialmente di nodi di input, che, a





 **Fig. 1** - Assistenti vocali di Amazon.

fronte di una determinata configurazione, permettono di triggerare dei flussi tramite controllo vocale. Questa famiglia di nodi è perfetta per realizzare applicazioni domotiche in quanto permette di controllare luci, relè, interruttori, tapparelle elettriche e via dicendo. Il secondo gruppo di nodi permette invece di interagire con la moltitudine di APIs messe a disposizione dall'ecosistema Alexa, quindi sono più orientati alla gestione dei dispositivi Amazon da Node-RED. Con questa famiglia di nodi è possibile quindi controllare Alexa come text-to-speech device (facendo quindi riprodurre vocalmente una stringa), o controllare la riproduzione di file musicali, oppure ancora inviare notifiche push ad un dispositivo. La lista è decisamente non esaustiva, dal momento che le operazioni eseguibili sono davvero tante, quindi non perdiamo tempo e passiamo alla descrizione di una libreria del primo gruppo, che ci consentirà di controllare tramite comandi vocali Node-RED e, di conseguenza, le interfacce hardware del dispositivo su cui lo stiamo facendo girare, ossia nel nostro caso, la nostra fida Raspberry Pi.

## NODE-RED COME ALEXA DEVICE

Come accennato precedentemente, esiste una famiglia di nodi che permette sostanzialmente di emulare dispositi-

tivi Alexa tramite Node-RED. Esistono diverse librerie in questa famiglia, ognuna con le sue caratteristiche peculiari ed i suoi pregi e difetti; noi ne consigliamo una che riteniamo decisamente valida e che si basa su una Alexa skill e su un servizio di Bridging per garantire l'interfacciamento.

Il nome della libreria è **node-red-contrib-alexa-home-skill** (realizzata da Ben Hardill), e può essere installata, come di consueto, cercandone il nome tramite la l'opzione "manage palette" di Node-RED.

Affinchè la libreria funzioni correttamente è necessario eseguire i seguenti passi:

1. Installare la libreria in Node-RED
2. Registrarsi al servizio di bridge con la skill Alexa
3. Aggiungere le descrizioni dei dispositivi che si desidera implementare
4. Installare la skill Node-RED sull'app Alexa

Iniziamo quindi installando la libreria, che è composta da due nodi più un configuration node (vedi Fig.2).

Descriviamo brevemente i nodi che compongono la libreria:


**alexa-home**: questo è il nodo base che permette di intercettare un comando vocale per uno specifico dispositivo. I dispositivi devono essere preventivamente configurati tramite il servizio di bridge con la skill, prima di poter essere utilizzati. La configurazione contiene, oltre al nome del dispositivo stesso, anche le sue proprietà, tra cui ad esempio la tipologia (lampada, interruttore, tapparella, etc..) ed i comandi vocali ai quali il dispositivo può obbedire.

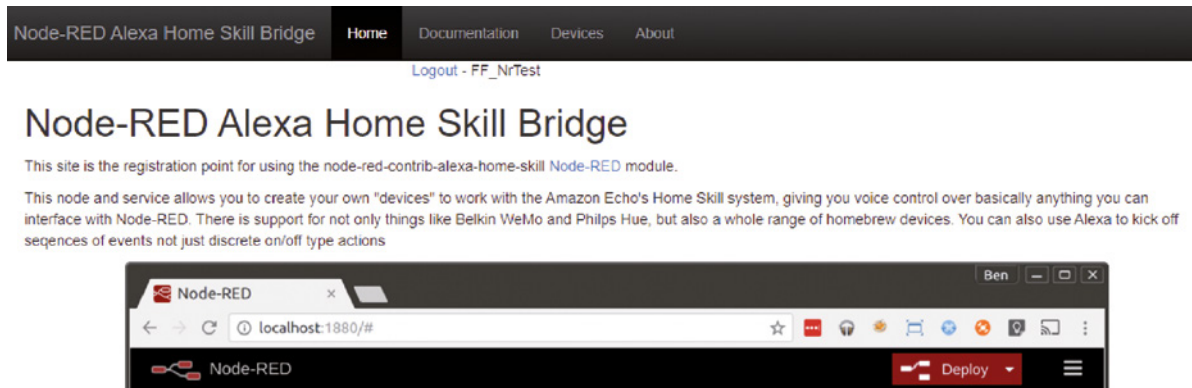
**alexa-home-resp**: questo nodo permette di inviare una risposta personalizzata in merito all'esecuzione di una determinata operazione (ad esempio per confermare che l'attuazione di una luce è andata a buon fine). Non ci soffermeremo molto su questo nodo, in quanto le medesime operazioni possono essere eseguite in maniera più semplice e completa, tramite alcuni nodi appartenenti al secondo gruppo.

**alexa-home-conf**: questo è il classico nodo di configurazione che permette di collegare un determinato account del servizio di bridge ad una specifica istanza di Node-RED, e quindi permettere di accedere ai dispositivi configurati per quello specifico account.

Una volta installata la libreria, prima di iniziare ad utilizzarla sarà necessario registrarsi al seguente sito web: <https://alexa-node-red.bm.hardill.me.uk/>, la cui home page è rappresentata in Fig.3. Una volta registrati sul sito possiamo iniziare ad aggiungere i dispositivi che desideriamo emulare tramite Node-RED. Come primo semplice esempio possiamo aggiungere un LED della nostra Raspberry Pi, che poi andremo ad attivare tramite un comando vocale. Spostiamoci quindi nella sezione "Devices" del sito



 **Fig. 2** - Nodi della libreria node-red-contrib-alexa-home-skill.



**Fig. 3** - Home page del servizio di bridge.

Add New Device

**Name:**

LED

**Description:**

This device will control a RPI LED.

**Actions**

On: ☒ Off: ☒

%: ☐ +%: ☐ -%: ☐

°C/F: ☐ +°C/F: ☐ -°C/F: ☐

**Fig. 4** - Inserimento di un nuovo dispositivo.

a cui ci siamo appena registrati e clicchiamo sul pulsante "Add Device" per configurare un nuovo dispositivo. Verrà aperta una semplice maschera di configurazione che ci permetterà di configurare il nostro device. Trattandosi di un semplicissimo esempio, possiamo inserire la configurazione riportata in **Fig. 4**.

Al termine dell'operazione, se tutto è andato a buon fine, il dispositivo comparirà nella lista dei dispositivi, come riportato in **Fig. 5**. A questo punto, per far riconoscere i dispositivi ad Alexa, è necessario installare una skill nell'app Alexa (o tramite il vostro account Amazon) e collegare l'account del servizio di bridging in cui abbiamo configurato il nostro dispositivo. La skill da installare si chiama Node-RED (di Ben Hardill), ed una immagine della pagina descrittiva della skill è visibile in **Fig. 6**.

La skill si può facilmente trovare cercando la voce "Node-RED" dall'app Alexa o dal vostro account Amazon. Una volta installata la skill va attivata e, dopo l'attivazione, verranno richieste le credenziali che avete inserito al momento della registrazione al servizio. A questo punto il dispositivo che è stato precedentemente aggiunto può essere riconosciuto da Alexa. Possiamo utilizzare il comando vocale "Alexa, trova nuovi dispositivi" per consen-

## Devices

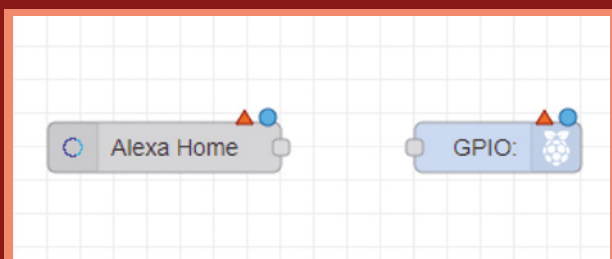
Name	Description	Actions
LED	This device will control a RPI LED.	<div> <div></div> <div></div> </div>
<div> <div>Edit</div> <div>Delete</div> </div>		
<div>Add Device</div>		

After making any changes remember to ask Alexa to update devices it knows about by saying "Alexa, discover devices"

**Fig. 5** - Lista dei dispositivi.



**Fig. 6** - Skill Alexa per l'interfacciamento con il servizio di Bridge.



**Fig. 7** - Inserimento di un nodo alexa-home e di un nodo rpi-gpio-out.

tire l'associazione. Dopo qualche istante Alexa dovrebbe confermare che un dispositivo denominato LED è stato trovato ed associato.

Possiamo tornare su Node-RED ed andare avanti con la creazione di un semplice flow per testare il nuovo dispositivo. Nella palette cerchiamo i nodi della categoria Alexa ed aggiungiamo un nodo alexa-home, oltre ad un nodo rpi-gpio-out, come illustrato in Fig.7.

Configuriamo, come di consueto, il nodo rpi-gpio-out per controllare un GPIO, e impostiamolo per controllare il GPIO 18, che, usando lo Shield Raspberry I/O

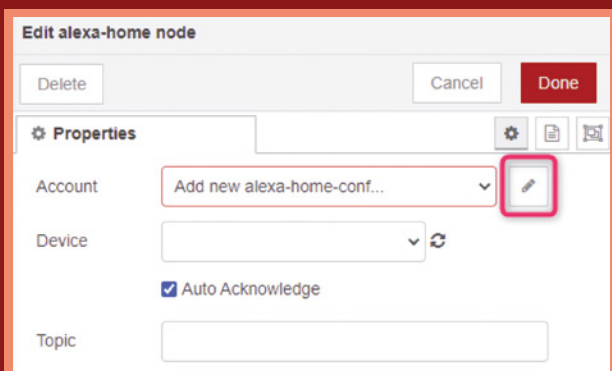
(7100-FT1060M), è collegato al primo LED.

Passiamo ora alla configurazione del nodo alexa-home, facendo un doppio click sul nodo stesso per accedere alla finestra delle proprietà. Qui clicchiamo sull'icona a forma di matita per accedere alla configurazione del configuration node alexa-home-conf, come illustrato in Fig. 8.

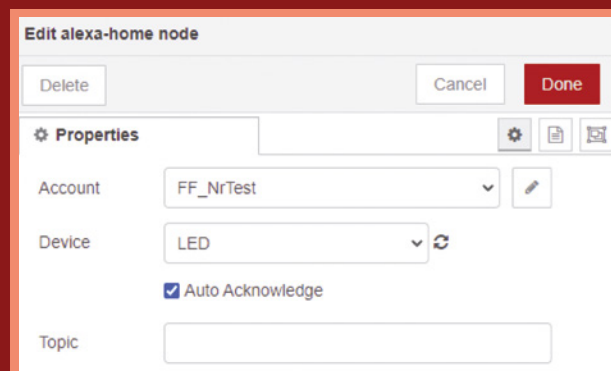
Il configuration node richiede semplicemente di inserire le credenziali dell'account creato in precedenza, quindi eseguiamo e clicchiamo su "Update". Una volta tornati sulla finestra delle proprietà del nodo alexa-home clicchiamo su "Done" ed eseguiamo un deploy. A questo punto, aprendo nuovamente la finestra delle proprietà, dovremmo poter vedere i nostri dispositivi (in questo caso, se non ne sono stati aggiunti altri, il dispositivo LED) sul drop-down control "Device". Selezioniamo LED, mettiamo il segno di spunta su Auto Acknowledge e clicchiamo nuovamente su "Done", come illustrato in Fig. 9.

A questo punto possiamo collegare con un wire l'uscita del nodo alexa-home e l'ingresso del nodo rpi-gpio-out (Fig.10) e testare il nostro flow.

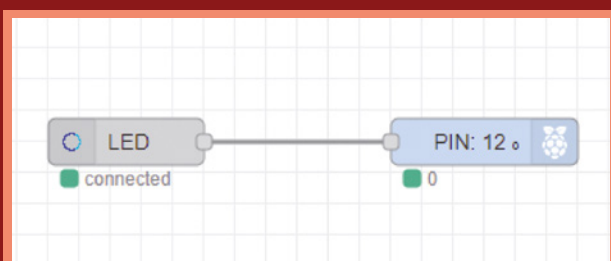
Se tutto è stato fatto correttamente, usando il comando vocale "Alexa, accendi LED" si dovrebbe accendere il LED sullo shield, mentre con il comando "Alexa, spegni



**Fig. 8** - Accesso al configuration node.



**Fig. 9** - Configurazione del nodo alexa-home.



**Fig. 10** - Flow per il controllo di un LED tramite comando vocale con Alexa.

LED" il LED dovrebbe spegnersi. Dal momento che è stata utilizzata l'opzione "Auto Acknowledge", Alexa fornirà automaticamente un feedback sul risultato dell'operazione ("OK" se tutto è andato per il verso giusto o un qualche tipo di errore nel caso contrario). Come avrete notato non è necessario nessun tipo di adattamento del messaggio di uscita in questo caso, in quanto il nodo alexa-home restituisce un boolean su msg.payload che riporta lo stato booleano associato al comando vocale impartito (true nel caso di "accendi LED" e false nel caso di "Spegni LED"). Naturalmente nel caso di attuazioni più complesse rispet-



**Fig. 11** - Nodi della libreria node-red-contrib-alexa-remote2-applestrudel.

to ad un semplice LED potrebbe essere necessario fare qualche adattamento al messaggio in uscita da alexa-home, ma sostanzialmente partendo da questa semplice base è possibile controllare (almeno con dei comandi di base) virtualmente qualsiasi dispositivo interfacciabile con Node-RED.

## INTERFACCIAMENTO CON ALEXA API

Ora che abbiamo sperimentato con il primo gruppo di nodi, passiamo ad analizzare il secondo, ossia i nodi che ci consentono di interfacciarci con le API di Alexa. In questa famiglia di nodi, il rappresentante più noto è sicuramente la libreria node-red-contrib-alexa-remote2-applestrudel, che ha subito vari fork, passando da essere chiamata alexa-remote and alexa-remote2, cakebed ed infine alesxa-remote2-applestrudel, che è la versione che vi raccomandiamo di installare. Una volta installata questa libreria mette a disposizione un ampio set di nodi, riportati in Fig. 11. Come di consueto facciamo una rapida carrellata, per poi passare a qualche dettaglio sulla configurazione e ad un esempio pratico.

**alexa-remote-init:** nodo utilizzato per effettuare il controllo manuale sull'inizializzazione. Non è strettamente necessario utilizzarlo in quanto esiste una opzione di auto-init nel configuration node.

**alexa-remote-event:** nodo che permette di intercettare eventi, come errori, modifiche ad alcuni settaggi come il volume ed altro ancora. Non tutti gli eventi vengono però notificati da questo nodo.

**alexa-remote-smarthome:** nodo per il controllo dei dispositivi di tipo casa intelligente. Questo nodo permette sia di controllare dispositivi standard, come lampadine intelligenti e simili, sia i dispositivi emulati da Node-RED (come il LED dell'esempio precedente).

**alexa-remote-routine:** questo nodo permette di triggerare vari tipi di routine, tra cui:

- **Speak:** riproduce una stringa di testo.
- **Speak at volume:** riproduce una stringa di testo ad uno specifico volume.
- **Text Command:** permette di inviare tramite una stringa un comando vocale.
- **Wait, Stop:** utilizzati per il controllo di esecuzione nelle sequenze.
- **Prompt, Phrase:** permette di far eseguire ad Alexa propt o frasi standard (ad esempio fornire notizie sul tempo o sul traffico in tempo reale o utilizzare formule di saluto standard).
- **Sound:** riproduce un suono (ad esempio un allarme o il verso di un animale).
- **Volume:** permette di controllare il volume.
- **Music:** Permette di riprodurre musica.
- **SmartHome:** permette di accedere al controllo dei dispositivi per la casa intelligente (sostanzialmente la stessa funzione del nodo alexa-remote-smarthome).



- **Launch Skill:** permette di lanciare una skill di Alexa (naturalmente la skill dev'essere installata).
- **Routine:** permette di eseguire una routine predefinita in Alexa.
- **Push Notification:** permette di inviare una notifica push al dispositivo associato all'account.
- **Node:** permette di implementare una sequenza complessa contenente, in numero ed ordine arbitrario, le funzioni elencate in precedenza.

**alexa-remote-echo:** questo nodo permette di gestire alcune funzioni fondamentali dei dispositivi, sia amministrative che non (come ad esempio rinominare o cancellare un dispositivo o gestire il pairing o la connessione di dispositivi bluetooth)

**alexa-remote-other:** questo nodo fornisce un'interfaccia con tutte le altre funzioni (ad esempio varie funzioni per il reperimento di informazioni o la gestione delle notifiche).

**alexa-remote-list:** questo nodo fornisce un'interfaccia dedicata per la gestione delle liste.

**alexa-remote-account:** questo è il configuration node che permette di interfacciare il nostro account Amazon con Node-RED ed impostare alcune configurazioni di base.

Ora che abbiamo dato uno sguardo ai vari nodi messi a disposizione, passiamo alla configurazione e ad un semplice esempio. Per cominciare trasciniamo sul workspace un nodo "Routine" e facciamo doppio click per accedere alla finestra delle proprietà. Come di consueto, la prima cosa da fare sarà impostare le proprietà del configuration node, quindi clicchiamo sull'icona a forma di matita (Fig.12). Qui impostiamo:

**Nome:** il nome che preferite per la vostra configurazione (nell'esempio MyNrConfig).

**Auth method:** Proxy

**This IP:** inserite l'IP locale della Raspberry sulla quale state facendo girare Node-RED (nell'esempio 192.168.1.10)

**Port:** lasciate pure 3456

**File Path:** qui va indicato il path di un file di testo locale nel quale verranno salvate le credenziali per gli accessi successivi

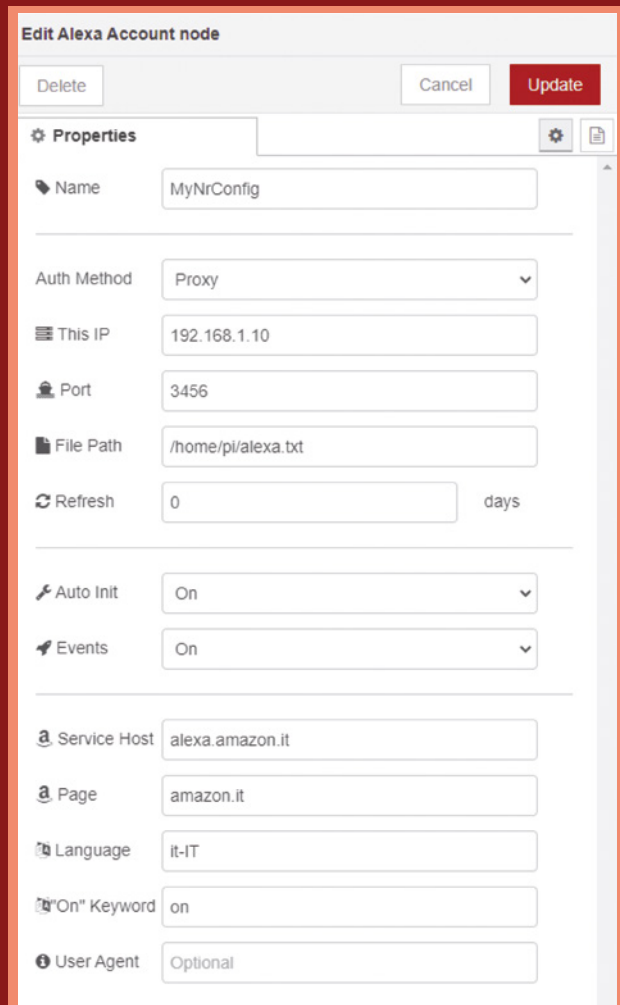
**Refresh:** 0

**Service Host:** alexa.amazon.it

**Page:** amazon.it

**Language:** it-IT

Per il resto lasciate sempre i valori di default. Salvate ed eseguite un deploy. A questo punto dovrete notare che sotto al nodo routine che avevate trascinato sul workspace compare il seguente messaggio: "open 192.168.1.10:3456 in your browser" (Fig.13). Naturalmente l'indirizzo IP sarà quello che avete inserito in fase di configurazione, ossia quello della vostra Raspberry nella vostra rete locale. Se aprite il browser all'indirizzo indicato dovrete trovare una pagina di autenticazione sulla quale dovrete inserire le

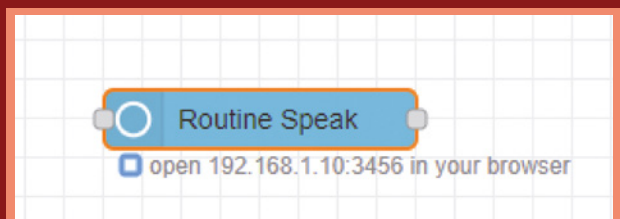


**Fig. 12** - Finestra di configurazione del configuration node di alexa remote2 applestrudfel.

credenziali del vostro account amazon.

Se tutto viene eseguito correttamente dovrete visualizzare un messaggio sul browser che conferma il successo dell'operazione e visualizzare la scritta "ready" sotto il nodo routine. A questo punto i nodi alexa remote sono operativi e possono essere utilizzati.

Per testarli possiamo utilizzare lo stesso esempio utilizzato



**Fig. 13** - Messaggio con l'indirizzo IP della pagina di autenticazione.

## Intervista a Nick O'Leary

Nick O'Leary è uno dei creatori di Node-RED, lo abbiamo raggiunto via mail per fargli qualche domanda sul passato, presente e futuro del tool che ha creato.

**Elettronica In:** Ciao Nick, come puoi immaginare molti dei nostri lettori sono curiosi riguardo a come lo sviluppo di Node-RED ha avuto inizio. Puoi darci qualche dettaglio a riguardo?

**Nick:** Certamente. Nel 2013 stavo lavorando presso l'IBM Emerging Technology Group. A quel tempo lavoravo su progetti per diversi clienti, spesso mettendo insieme diversi elementi tecnologici per dare vita ai progetti più disparati. Come parte del ruolo eravamo attivamente incoraggiati a spendere parte del nostro tempo usando nuove tecnologie e mettendo insieme proof-of-concepts di nuove idee. Avendo un background in MQTT ho realizzato una semplice demo di un tool che permetteva di connettere insieme nodi che rappresentavano topics MQTT per mezzo di "wires". Non c'è voluto molto ad aggiungere un pulsante che permetteva di generare il codice in grado di implementare il sistema MQTT progettato. A quel punto un collega che stava per iniziare un nuovo progetto per un cliente mi suggerì che sarebbe stato molto utile avere la possibilità di mappare messaggi MQTT su una porta seriale, e così il gruppo di nodi Serial venne aggiunto al PoC. Da quel punto in avanti ogni nuovo progetto aggiunse un nuovo set di requisiti e un nuovo set di nodi al tool. In breve diventò chiaro che non si trattava più solo di MQTT, ma avevamo tra le mani un tool per la progettazione dei sistemi più disparati. La prima versione di Node-RED era nata a quel punto.

**Elettronica In:** Avresti mai immaginato che Node-RED potesse avere un simile successo?

**Nick:** Nei primi mesi siamo stati molto impegnati a consegnare progetti per i clienti, spesso usando Node-RED per realizzarli. Sapevamo di avere qualcosa di utile tra le mani, ma la vera domanda era cosa fare con quel qualcosa. Sin dall'inizio abbiamo deciso che la migliore cosa da fare fosse di renderlo open-source, per far sì che la vasta community degli sviluppatori IoT potesse metterci le mani sopra. Dopo essere passati per il processo interno della compagnia abbiamo ricevuto l'approvazione a renderlo open-source e così il progetto Node-RED su GitHub vide la luce. Non avrei mai potuto immaginare quando la community potesse trovarlo utile e quanto potesse farlo crescere, ma è successo. Continuo a volte a vederlo come il mio piccolo side-project, ma da molto tempo ormai è molto più di quello.

**Elettronica In:** Quali sono i tuoi programmi adesso? Abbiamo visto che hai lasciato IBM, lo hai fatto perché vuoi dedicarti al



100% a Node-RED nel prossimo futuro?

**Nick:** Sono stato incredibilmente fortunato a trovare un ruolo in IBM, che mi ha concesso di guidare il progetto di Node-RED e renderlo open-source. Avrei potuto continuare felicemente in quel ruolo, ma all'inizio del 2021 mi è stata data l'opportunità di costruire qualcosa di più grande con Node-RED. Quindi in Aprile ho lasciato IBM ed ho fondato la mia compagnia, FlowForge Inc., che ha lo scopo di costruire una piattaforma attorno a Node-RED, così che possa essere più facile per altre compagnie adottarlo.

**Elettronica In:** Quali compagnie fanno uso di Node-RED?

**Nick:** C'è una crescente lista di compagnie che hanno adottato Node-RED. E' molto interessante osservare da quali settori provengono. Molte di loro sono società che lavorano nell'IoT, come ci si potrebbe aspettare, ma ci sono altre compagnie che lo usano e che lavorano in altri settori, come l'healthcare, l'insurance e l'automotive.

**Elettronica In:** Quali sono le funzionalità più interessanti della release corrente (Node-Red 2.0)? E quali sono le più promettenti feature future che puoi anticiparci?

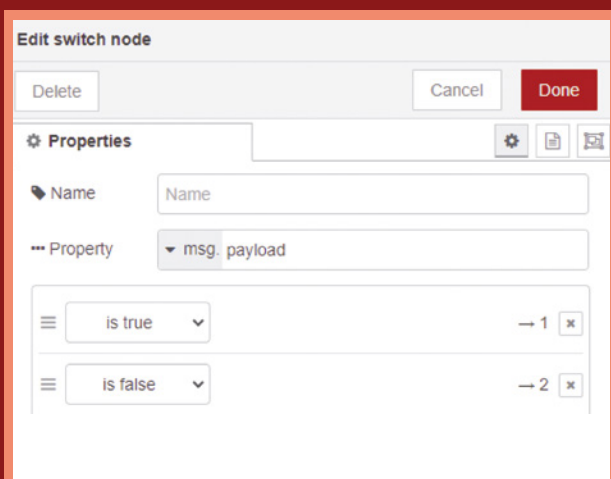
**Nick:** Nella release 2.0 ci sono due nuovi tools piuttosto interessanti: un flow debugger ed un Flow Linter.

Il debugger permette di inserire breakpoints nei flows, esattamente come si farebbe con un tradizionale debugger a livello di codice. Il Linter è un analizzatore che controlla il flow usando una serie di regole e permette di evidenziare potenziali problemi nel flow. Siamo ancora agli albori per entrambi questi tools, ma in futuro forniranno agli utilizzatori nuovi superpoteri per aiutarli a creare applicazioni con un elevato livello di qualità. Un nuovo elemento sul quale abbiamo iniziato a lavorare è un Flow Testing Framework. Quello che vorremmo fare è fornire agli sviluppatori uno strumento che gli consenta di creare facilmente test per i loro flow, in modo da garantire che funzionino come ci si aspetta.

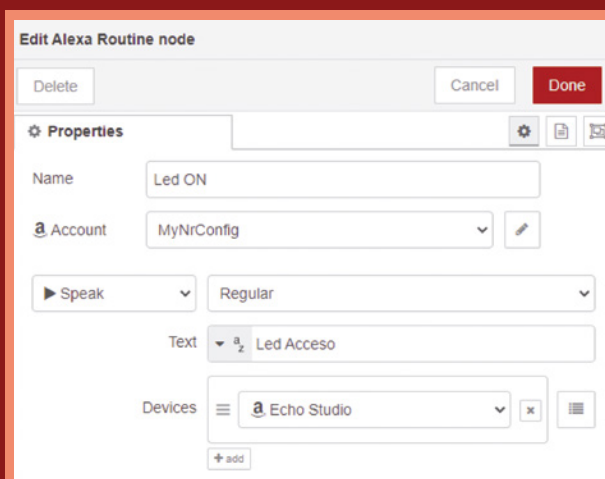
in precedenza, completandolo con una conferma customizzata alle operazioni di accensione e spegnimento del LED. Per farlo trasciniamo sul workspace un altro nodo "Alexa Routine" ed un nodo Switch.

Configuriamo il nodo switch con due regole, rispettivamente "is true" ed "is false" (vedi Fig.14), per creare due

output alternativi che vengono triggerati in funzione dell'operazione richiesta (ricordate che il nodo alexa-home ritorna appunto un boolean come risultato della richiesta di accensione o spegnimento del LED). Ora configuriamo i due nodi Alexa routine per riprodurre sul dispositivo che vogliamo rispettivamente le due frasi "LED Acceso" e "LED



**Fig. 14** - Configurazione del nodo Switch.

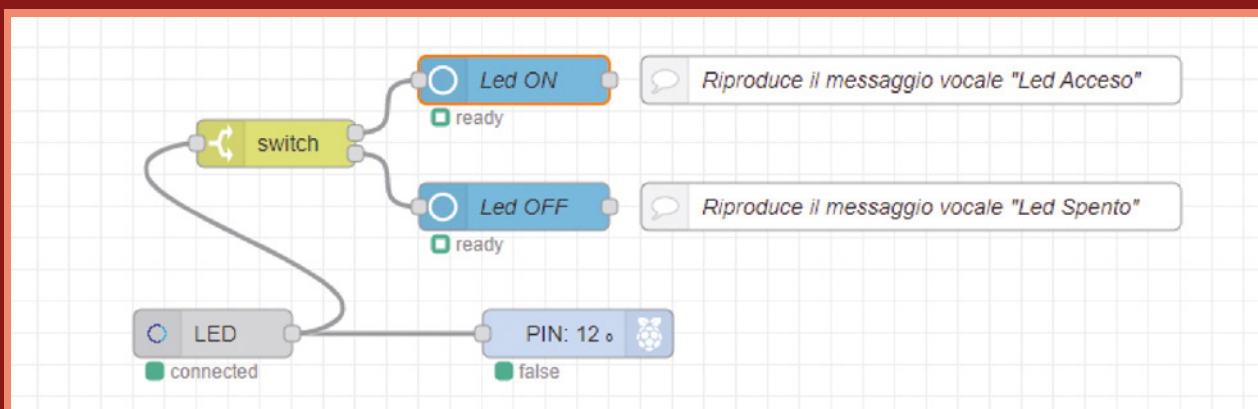


**Fig. 15** - Configurazione del nodo Alexa Routine usato per riprodurre il comando vocale "LED acceso".

spento" e colleghiamoli ai due rami "is true" ed "is false" del nodo switch. Ovviamente il nodo che riproduce la frase "LED acceso" sul ramo "is true" e l'altro su "is false". Infine colleghiamo l'uscita del nodo alexa-home all'ingresso del nodo switch. Infine rimuoviamo la spunta sulla checkbox "Auto Acknowledge" del nodo alexa-home. In Fig.15 è riportata la finestra delle proprietà di uno dei nodi Alexa Routine ed in Fig.16 un'immagine del flusso completo. A questo punto se ritestate il vostro flow noterete che le risposte di Alexa sono cambiate, infatti invece di rispondere sempre "OK" (funzione di auto acknowledge, che abbiamo disabilitato), risponde con le frasi "LED acceso" e "LED spento" a conferma delle due operazioni. Come avrete potuto facilmente notare, quelli che abbiamo presentato sono solo due semplicissimi esempi delle potenzialità derivanti dalla combinazione di Node-RED ed Alexa, lasciamo ai lettori più interessati ed intraprendenti l'esplorazione delle varie funzioni aggiuntive.

## CONCLUSIONI

Siamo arrivati alla conclusione del corso Node-RED. Durante questa avventura, dopo aver gettato le necessarie basi teoriche sulla FBP (Flow Based programming), siamo passati dai primi, semplici, esempi di gestione dell'I/O (GPIO, I2C, UART, 1-wire), ad esempi sempre più complessi, come la connettività (TCP, UDP, http, WebSocket ed MQTT), l'interfacciamento con le app di messaggistica (Whatsapp e Telegram) ed infine l'integrazione con gli assistenti vocali. Ci sarebbero molti altri temi, che per questioni di tempo, non possono essere trattati in questo corso, come il Machine Learning e l'AI, l'interfacciamento con i social (Facebook, Twitter) ed altro ancora, insomma Node-RED è uno strumento potente e flessibile, ma allo stesso tempo, alla portata di tutti, quindi non vi resta altro da fare che prendere una Raspberry (se non ne siete già in possesso) ed iniziare a sviluppare i vostri flow e le vostre automazioni. □



**Fig. 16** - Flusso completo con risposte customizzate.