

CONOSCERE E USARE

di FRANCESCO FICILI

4

Node-RED

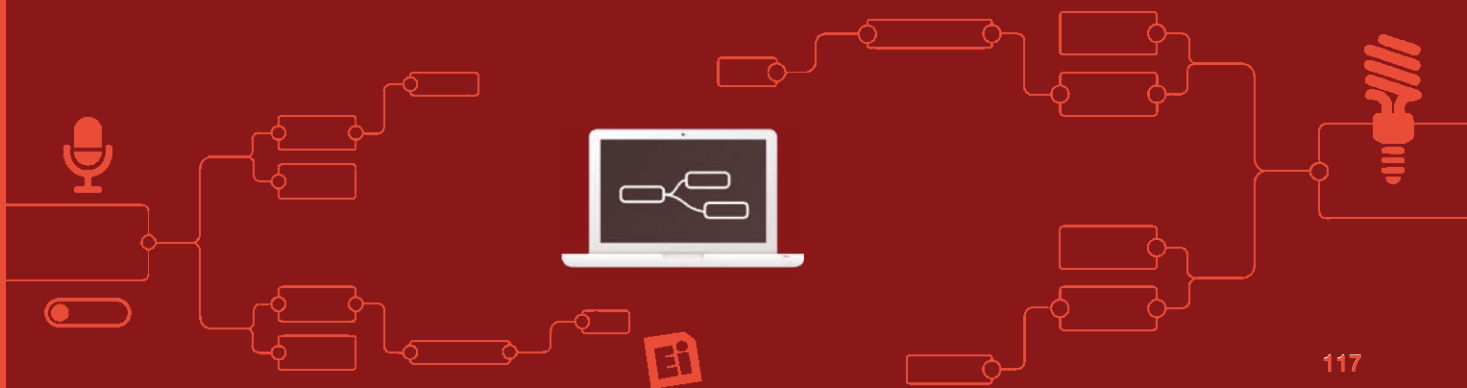
Iniziamo a scoprire Node-RED, un tool di flow-based programming, orientato all'IoT ed alla connettività, originariamente sviluppato dall'IBM Emerging Technology Services team e adesso parte della JS Foundation. In questa puntata ci occupiamo dell'implementazione della parte grafica delle nostre applicazioni, sfruttando la Node-RED dashboard, uno dei pacchetti di nodi aggiuntivi maggiormente utilizzati. Quarta Puntata.

Nella precedente puntata abbiamo iniziato a prendere confidenza con la gestione delle periferiche hardware tramite Node-RED, utilizzando, come di consueto per questo corso, la Raspberry Pi come piattaforma hardware. Nello sviluppo di applicazioni IoT, o comunque orientate alla connettività, l'utilizzo delle periferiche è sicuramente una faccia della medaglia, in quanto è fondamentale per l'interfacciamento dello specifico sensore o attuatore che si desidera rendere "smart" o comunque remotizzare, in modo che sia gestibile attraverso una opportuna interfaccia. La realizzazione della suddetta interfaccia è l'altro lato della medaglia e può essere declinata in diverse forme. Sicuramente una delle forme più comunemente utilizzate è quella grafica, nella modalità in cui l'istanza di Node-RED sulla nostra Raspberry Pi agisce come webserver, fornendo l'interfaccia grafica quando ci colleghiamo tramite

un browser ad uno specifico indirizzo IP. Node-RED ci viene incontro in questa specifica modalità di utilizzo con un componente estremamente interessante: la dashboard. Tuttavia, prima di vedere di cosa si tratta nel dettaglio, dobbiamo affrontare il tema della configurazione, in particolare perché, come vedremo in seguito, alcune modalità di utilizzo dei nostri dispositivi ci espongono a seri rischi di Cybersecurity.

CONFIGURARE NODE-RED

Come abbiamo detto nella parte introduttiva, in questa puntata porremo molto l'accento sulla creazione di interfacce utente remote, tramite Node-RED, utilizzando un componente (la dashboard) che è in grado di esporre parti grafiche interattive su una specifica pagina web, creando di fatto delle webapp. Questa modalità presenta notevoli vantaggi, tra cui la semplicità di utilizzo



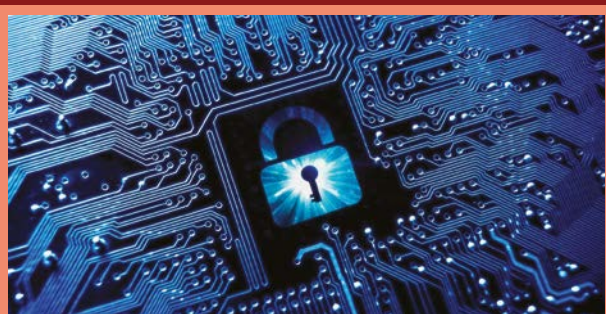


 Fig. 1 - Cybersecurity.

e l'estrema compatibilità. Infatti per gestire uno specifico dispositivo remoto sarà sufficiente collegarci all'indirizzo IP in cui Node-RED espone il webserver e non è importante con quale dispositivo (la webapp è compatibile con i vari smartphone in circolazione, sia Android che iOS, che anche con i vari browser utilizzati in ambiente desktop). Tuttavia se vogliamo remotizzare un'applicazione di questo tipo, uscendo dai confini della nostra rete locale, non abbiamo altra scelta se non quella di aprire una porta del nostro router. Questa azione, inevitabilmente, ci sottopone ad un rischio: se non adeguatamente protetta, la nostra interfaccia web, come anche l'editor di Node-RED, potrebbero essere sfruttati da una terza parte a scopi malevoli. Dobbiamo quindi affrontare un problema di cybersecurity.

Fortunatamente Node-RED dispone di tutta una serie di predisposizioni che, se opportunamente attivate, ci consentiranno di esporre i nostri pannelli di controllo al di fuori della nostra rete locale con la ragionevole sicurezza di poter resistere ad attacchi esterni (in cybersecurity non si parla mai di sicurezza assoluta, in quanto avendo un determinato quantitativo di tempo a disposizione, anche le interfacce più sicure possono essere forzate). Prima di addentrarci però nei dettagli relativi alla protezione della nostra istanza di Node-RED è opportuno analizzare come Node-RED affronta il problema della configurazione, che consente anche di attivare le varie features di Cybersecurity integrate.

Node-RED, al suo avvio, carica una serie di opzioni di configurazione presenti sul file `settings.js`, che si trova di default nella sua directory principale. Di conseguenza, per configurare il comportamento della nostra istanza di Node-RED possiamo semplicemente intervenire su questo file, attivando o modificando le varie opzioni di configurazione che sono presenti. Le opzioni disponibili sono diverse ed impattano su innumerevoli aspetti relativi all'editor, alla gestione dei files ed al comportamento in generale di Node-RED.

La maggior parte delle opzioni sono commentate e corredate di una breve descrizione, per attivarle è quindi sufficiente decommentare la sezione specifica e fare le modifiche del caso. Ad esempio, se volessimo visualizzare l'editor all'interno di uno specifico path, come `"/editor"`, basterebbe

attivare l'opzione `httpAdminRoot`, e modificarla in:

```
httpAdminRoot: '/editor',
```

Riavviando Node-RED (può essere consigliabile riavviare del tutto la Raspberry Pi, tramite il comando `sudo reboot`), non visualizzeremo più il nostro editor su `<localhost>:1880`, ma su `<localhost>:1880/editor`.

Il dettaglio delle varie opzioni di configurazione esula dagli obiettivi di questo corso, ma una descrizione dettagliata delle varie configurazioni possibili può essere reperita dal sito web di Node-RED al seguente link:

<https://nodered.org/docs/user-guide/runtime/configuration>.

OPZIONI DI SECURITY

Ora che abbiamo capito come fare per configurare opportunamente Node-RED, entriamo nel dettaglio delle configurazioni di Cybersecurity che vogliamo mettere in atto per proteggere la nostra istanza di Node-RED nel momento in cui volessimo renderla accessibile dall'esterno della nostra rete locale.

La prima cosa che vogliamo fare è proteggere l'editor, per fare in modo che solo gli utenti autorizzati possano accedere e fare modifiche ai flows. Per garantire questo tipo di protezione di accesso, Node-RED fornisce due sistemi:

- Autenticazione basta su username/password
- Autenticazione basata su OAuth/OpenID providers (come ad esempio Twitter o GitHub)

Noi ci concentreremo solo sulla prima modalità in questa puntata. Per creare utenti con relative credenziali di accesso dobbiamo decommentare e modificare la proprietà `adminAuth` nel file `settings.js` dell'istanza di Node-RED che vogliamo proteggere. La proprietà `adminAuth` dispone di due campi:

- `Type`: rappresenta il tipo di protezione che vogliamo implementare. La modalità username/password è definita "credentials"
- `Users`: è un array di javascript objects che rappresenta i vari utenti che vogliamo creare, con relative password e diritti di accesso.

Quindi, se vogliamo creare due utenti, un utente `admin`, con diritti di lettura e scrittura e un utente `guest` con diritti di sola lettura, dovremo modificare la proprietà come illustrato nel **Listato 1**.

Come si può vedere dal listato le password sono opportunamente crittografate tramite l'algoritmo di hashing `bcrypt`, in modo da non essere conservate in chiaro (nell'esempio gli hash delle due password sono stati ridotti per ragioni di spazio). Per generare gli hash delle password dei nostri utenti possiamo utilizzare direttamente la funzione di hash `bcrypt` integrata in Node-RED (`node-red admin hash-pw`), oppure utilizzare uno dei tanti servizi presenti online, come



Listato 1 – Creazione degli utenti admin e guest

```
adminAuth: {
  type: "credentials",
  users: [
    {
      username: "admin",
      password: "$2b$10$uf4IHPnu01B04pR...",
      permissions: "*"
    },
    {
      username: "guest",
      password: "$2b$10$D4vDDId7yb42kme1...",
      permissions: "read"
    }
  ]
},
```

ad esempio <https://passwordhashing.com/BCrypt>. Una volta creati gli utenti ed assegnati gli hash delle password non ci resta che assegnare i privilegi di accesso ad ogni utente. Nell'esempio di **Listato 1**, all'utente admin sono stati assegnati privilegi illimitati (*), mentre all'utente guest sono stati assegnati privilegi di sola lettura (read). Una volta fatta questa modifica e riavviato Node-RED, se proviamo ad accedere all'editor, verremo re-indirizzati su una pagina di login come quella riportata in **Fig. 2**, e solo inserendo una coppia di credenziali valide potremo effettivamente accedere.

Il meccanismo di autenticazione di Node-RED prevede la creazione di token di accesso, che hanno validità per un certo periodo. Di conseguenza non saremo costretti a re-inserire la password di accesso finché il token rimane valido, e il periodo di validità dei token può essere modificato agendo sulla proprietà `sessionExpiryTime` (il valore di default è di 7 giorni). Si faccia attenzione però a questo settaggio: periodi troppo lunghi potrebbero esporre a rischi legati alla sicurezza, mentre periodi troppo corti comportano la necessità di dover reinserire la password di accesso ogni volta che l'editor lo richiede (anche a runtime, mentre stiamo modificando i nostri flows).

Il meccanismo di protezione che abbiamo creato consente di bloccare l'accesso al nostro editor da parte di soggetti non autorizzati, ma è ancora vulnerabile, perché lo scambio di informazioni avviene su un canale HTTP non protetto. Di conseguenza le nostre credenziali di accesso sono trasmesse in chiaro dal client (il browser che stiamo utilizzando) verso il nostro server Node-RED, e di conseguenza potrebbero essere intercettate. Per eliminare questo rischio dob-

Listato 2 – Configurazione del canale HTTPS

```
https: {
  key: require("fs").readFileSync('/home/pi/.node-red/privatekey.pem'),
  cert: require("fs").readFileSync('/home/pi/.node-red/certificate.pem')
},
```

biamo passare su un canale sicuro di tipo HTTPS. Per farlo è sufficiente decommentare la proprietà `https` presente sempre sul nostro settings file. Questa proprietà contiene due campi, `key` e `cert`, che rappresentano la chiave privata e la catena di certificati necessari per la creazione del canale HTTPS. I campi `key` e `cert` non sono altro che i path della chiave privata e del certificato, in formato .pem. Quindi, supponendo di avere una coppia chiave/certificato valida, per re-indirizzare la connessione al nostro server Node-RED su HTTPS dobbiamo inserire una configurazione simile a quella riportata nel **Listato 2**.

Per la creazione della chiave privata e del certificato, è possibile seguire questa guida:

<http://stevesnoderedguide.com/securing-node-red-ssl>.

A questo punto, per raggiungere il nostro editor, non potremo più utilizzare una connessione HTTP, ma dovremo collegarci in HTTPS. Può essere comodo attivare anche la seguente opzione, sempre sul nostro file `settings.js`:

```
requireHttps: true,
```

che reindirizza le connessioni HTTP su HTTPS automaticamente.

A questo punto il nostro editor è ben protetto, e possiamo quindi decidere di esporlo verso l'esterno. Si faccia comunque sempre molta attenzione a questa operazione, dal momento che aprendo una porta del nostro router verso l'esterno ci esponiamo comunque ad attacchi, quindi si raccomanda sempre di evitare di inserire informazioni sensibili nei dispositivi a cui viene dato accesso e di proteggere sempre le varie interfacce che si espongono (come ad esempio SSH o simili).

NODE-RED DASHBOARD

Ora che la nostra istanza di Node-RED è protetta possiamo passare all'argomento principale di questa puntata, ossia, come accennato nella parte introduttiva, come realizzare interfacce grafiche per i nostri flows.

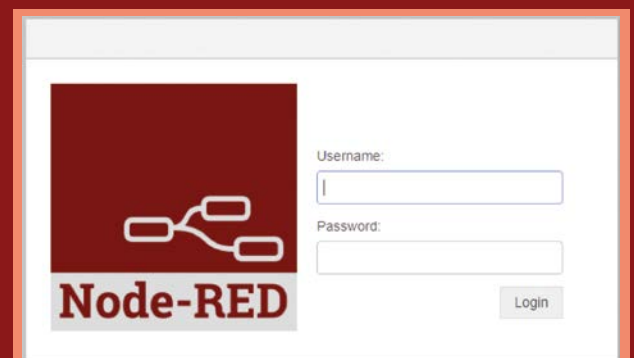


Fig. 2 - Protezione dell'accesso all'editor di Node-RED con username e password.

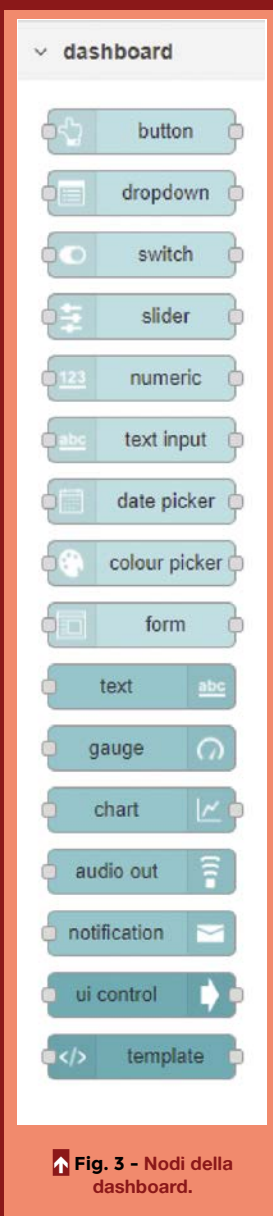


Fig. 3 - Nodi della dashboard.

Dal momento che Node-RED nasce come un ambiente di sviluppo con forti connotazioni IoT, è stato sin dall'inizio pensato un componente che permetta agli utilizzatori di realizzare semplicemente interfacce grafiche: la dashboard. La dashboard non è altro che un set di nodi (si installa esattamente come gli altri nodi e, una volta installato, compare in una sezione dedicata sulla palette) che permettono di inserire elementi grafici come pulsanti, gauges, grafici e molti altri, che ci permettono di visualizzare dati in real-time e di interagire con altri nodi presenti nei nostri flows. Per installare la dashboard è sufficiente cercare tramite l'opzione "Manage Palette" il componente node-red-dashboard. Una volta installato vedremo nella palette il

gruppo di nodi riportato in Fig. 3.

Oltre ai vari nodi viene installato anche un particolare pannello di gestione, denominato dashboard ed accessibile dalla sidebar (vedi Fig. 4), che ci permette di gestire layout, temi, font ed altre opzioni della parte grafica.

La dashboard è semplicissima da utilizzare, infatti, una volta installata, l'interfaccia grafica della nostra istanza di Node-RED risulta accessibile all'indirizzo:

<Node-RED server URL/IP>:1880\ui

Ossia dobbiamo semplicemente aggiungere "\ui" alla locazione a cui solitamente ci colleghiamo per accedere all'editor. Questo specifico path può anche essere modificato dall'utente, andando semplicemente a modificare la properties "ui" del file *settings.js*, in maniera del tutto simile a come abbiamo fatto per il path dell'editor nella parte iniziale di questa puntata (ricordiamoci sempre di riavviare Node-RED per rendere effettive le modifiche che facciamo al file di configurazione).

DASHBOARD NODES

Come abbiamo accennato in precedenza, la dashboard è composta da una serie di nodi, ed alla quasi totalità di questi nodi corrisponde un elemento grafico, denominato widget, con il quale sia il flow che l'utente possono interagire. E' possibile quindi visualizzare dati provenienti dal flow (ad esempio visualizzare la lettura di un sensore di temperatura su un grafico) oppure recepire input dall'utente ed eseguire azioni (ad esempio, attivare un relè alla pressione di un pulsante).

Facciamo quindi un'analisi di dettaglio dei vari widget che compongono la dashboard:

- **Audio Out:** questo widget permette di riprodurre audio (wav o mp3) o di sintetizzare vocalmente un testo passato come input.
- **Button:** questo widget crea un pulsante virtuale con il quale è possibile interagire. E' possibile impostare lo specifico *msg.payload* da inviare alla pressione del pulsante.
- **Chart:** questo widget crea un grafico, che può essere di vari tipi differenti (grafico a linee, a barre o a torta). Nel grafico vengono rappresentati i dati inviati alla porta di input del nodo, come ad esempio la lettura di un sensore.
- **Colour Picker:** questo widget crea un picker che permette di selezionare un colore.
- **Date Picker:** questo widget crea un picker che permette di selezionare una data.
- **Dropdown:** questo widget crea un dropdown menu e permette di selezionare una delle opzioni. Gli elementi del dropdown possono essere definiti dalla finestra di configurazione del nodo o anche tramite uno specifico campo (*msg.options*) inviato tramite un messaggio in ingresso.

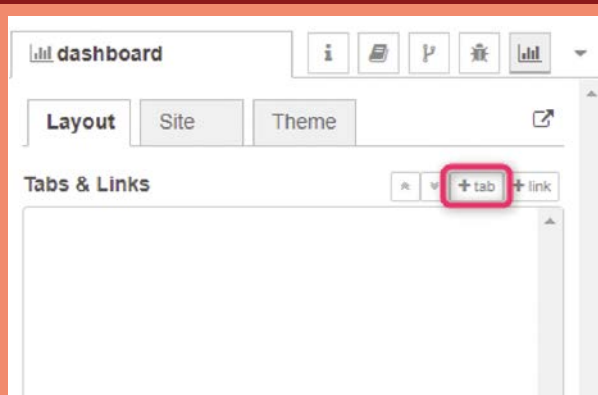


Fig. 5 - Aggiunta di un Tab.

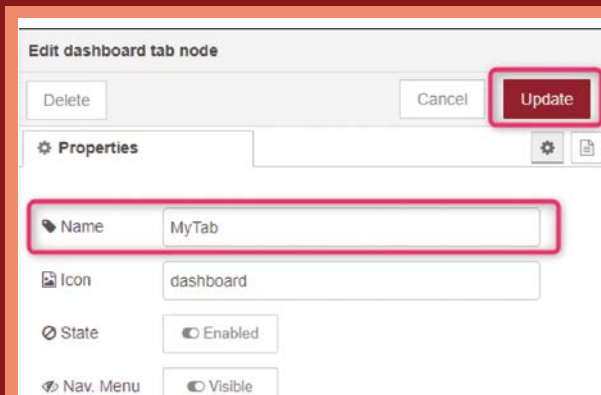


Fig. 6 - Rinominare il TAB.

- **Form:** questo widget crea un form i cui campi possono essere definiti tramite la finestra di configurazione del nodo.
- **Gauge:** questo widget crea un gauge che rappresenta il valore istantaneo passato alla porta di ingresso tramite *msg.payload*. Il nodo ha 4 modalità differenti per la gauge: standard, donut, compass e level.
- **Notification:** questo widget è in grado di creare notifiche per l'utente. La notifica può essere di tipo toast (popup che scompare dopo un tempo predefinito) oppure nella forma popup con pulsante di conferma (ed in questo caso può anche essere richiesto un input all'utente).
- **Numeric:** questo widget crea un input di tipo numerico.
- **Slider:** questo widget crea una slider orizzontale controllabile dall'utente.
- **Switch:** questo widget crea uno switch, ossia un pulsante che ha due posizioni stabili, a cui corrispondono due differenti valori di *msg.payload* inviati come output.
- **Text:** questo widget crea un elemento grafico testuale di output, e può quindi essere utilizzato come label. Il contenuto della label viene impostato tramite il valore di *msg.payload* inviato sulla porta in ingresso.
- **Text Input:** questo widget crea un elemento grafico testuale di input. Può supportare diverse modalità, come password, email, telephone e vari tipi di picker.
- **Template:** la dashboard ha anche una sua specifica tipologia di template node, che permette di creare widget personalizzati, tramite HTML e Javascript.
- **Ui control:** questo nodo permette di controllare dinamicamente la dashboard.

Una volta posizionato un nodo della dashboard sul nostro flow, dovremo occuparci di come il relativo widget viene posizionato nel layout visualizzato sulla UI. La dashboard è organizzata in elementi gerarchici che prendono il nome di Tabs e Groups. I Tabs sono gli elementi posizionati gerarchicamente più in alto e rappresentano appunto i vari tab che possono essere visualizzati dalla UI (ne può essere

visualizzato uno per volta). I groups invece sono contenuti nei tab e contengono al loro interno una serie di widgets. Di conseguenza, se vogliamo posizionare dei widget sulla nostra UI, dobbiamo creare almeno un tab ed un group. Questa operazione può essere eseguita sia dalla pagina di gestione della dashboard, che anche dalle singole pagine di configurazione dei vari nodi, nella sezione che permette di associare un nodo/widget ad un group. Nella pagina di gestione della dashboard è possibile anche gestire il layout dei gruppi all'interno dei tab e dei widget all'interno dei gruppi.

ESEMPIO

Vediamo a questo punto un piccolo esempio di come funziona la dashboard, in maniera da comprenderne meglio caratteristiche e funzionalità. Ipotizziamo di voler creare un semplice pannello con un controllo di tipo slider ed uno di tipo gauge, posizionati all'interno di un gruppo su un tab, e di voler controllare il valore visualizzato dal gauge tramite lo slider. Per prima cosa creiamo gruppo e tab, andando a selezionare la sezione dashboard della sidebar e posizionandoci sul tab Layout. Per aggiungere un tab clicchiamo sul pulsante "+ tab", come illustrato in Fig. 5. Una volta aggiunto il tab quest'ultimo comparirà nella sezione Tabs & Links; passandoci sopra con il mouse compariranno alla destra dell'elemento stesso tre opzioni: "+ group", "edit" e "link". Selezioniamo l'opzione edit per cambiare il nome al tab. Una volta selezionata l'opzione si aprirà una finestra delle proprietà dalla quale è possibile eseguire alcune operazioni, come rinominare il tab, aggiungere un'icona e altro. Cambiamo il nome da quello di default in MyTab e clicchiamo sul pulsante rosso Update per proseguire, come illustrato in Fig. 6.

A questo punto clicchiamo su "+ group" per aggiungere un nuovo gruppo.

Similmente a quanto visto nel caso del tab, anche i gruppi hanno una opzione edit, che possiamo usare per rinominare il gruppo, cambiandone il nome in "MyGroup".

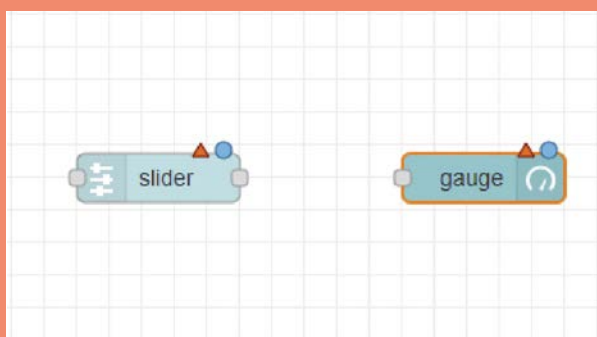


Fig. 7 - Posizionamento di un nodo slider e un nodo gauge sul workspace.

Anche in questo caso abbiamo varie opzioni che per il momento possiamo ignorare. A questo punto la nostra UI dispone di un tab e di un gruppo, quindi possiamo inserire i widget. Selezioniamo, tra i nodi della sezione dashboard, un nodo slider ed un nodo gauge e trasciniamoli sul workspace, come illustrato in **Fig. 7**.

Come si può vedere i nodi presentano le icone a forma di triangoli rossi, ad evidenziare un errore di configurazione (oltre al pallino blu ad indicare che ancora non è stato

eseguito il deploy). Questo perché i nodi non sono ancora stati esplicitamente assegnati a nessun gruppo. Questa assegnazione viene fatta dalla finestra di proprietà del nodo. Iniziamo con il nodo slider, cliccando 2 volte per accedere alla finestra properties. Da questa finestra assegniamo il gruppo tramite il controllo Group (primo dall'alto) e modifichiamo anche i seguenti 2 campi (come illustrato anche in **Fig. 8**):

Name: assegniamo come nome al controllo "MySlider".

Tooltip: aggiungiamo il seguente tooltip, "The slider will control the gauge".

Lasciamo i valori di default per il resto dei controlli e clicchiamo sul pulsante Done per confermare.

Continuiamo modificando le proprietà del nodo gauge, assegnando il medesimo gruppo ed assegnando "MyGauge" come nome al nodo, come illustrato in **Fig. 9**.

A questo punto, dato che vogliamo che lo slider controlli il gauge, colleghiamo con un wire l'output del nodo slider all'input del nodo gauge, ed eseguiamo infine il deploy. Per visualizzare l'interfaccia grafica creata, possiamo collegarci al path interno in cui è visualizzata la ui, ossia, come detto in precedenza:

```
<Node-RED server URL/IP>:1880/ui
```

Dove <Node-RED server URL/IP> rappresenta l'indirizzo della nostra raspberry. In **Fig. 10** è riportato un esempio di output di questo semplice flow con interfaccia grafica.

SECURITY DELLA UI

Ora che abbiamo visto come creare una interfaccia grafica con Node-RED è anche opportuno ritornare nuovamente sul tema della security, in quanto l'opzione di protezione vista all'inizio della puntata è valida solo per l'editor, ma non per la UI. Node-RED ha un secondo set di opzioni per proteggere il pannello della dashboard ed i contenuti statici (come le immagini), se ce ne sono.

Le due opzioni, che potete trovare sempre sul file *settings.js*, sono riportate nel **Listato 3**.

Anche in questo caso si tratta di una coppia username-password, e la password è da inserire esattamente nelle stesse modalità della password per l'accesso all'editor, ossia crittografata con algoritmo bcrypt.

ESEMPIO PRATICO: STAZIONE METEO CON BME280

Passiamo a questo punto all'esempio pratico di questa

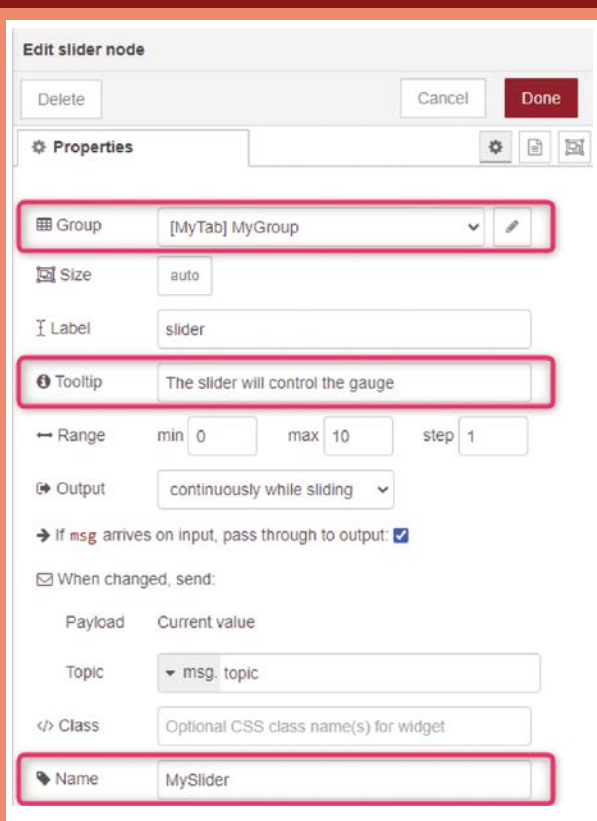


Fig. 8 - Proprietà del nodo slider.

❏ Listato 3 - Opzioni di protezione della dashboard e dei contenuti statici

```
httpNodeAuth: {user:"admin",pass:"$2b$10$uf4IHPnu01B04pR...","},
httpStaticAuth: {user:"admin",pass:"$2b$10$uf4IHPnu01B04pR...","},
```

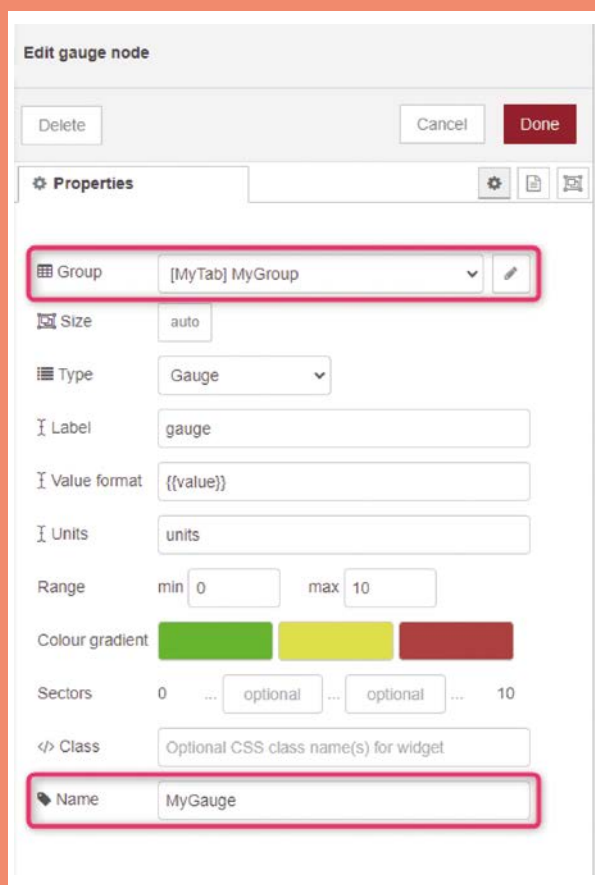


Fig. 9 - Proprietà del nodo gauge.

puntata: quello che ci proponiamo di ottenere è di realizzare una semplice stazione meteo che prenda dati di un sensore di temperatura, umidità e pressione e li visualizzi su una serie di gauge su una pagina web. Come hardware utilizzeremo ovviamente una Raspberry Pi ed il sensore BME280. Tale sensore può essere acquistato presso lo shop di futura elettronica ed ha come codice prodotto la sigla 1606-GY-BME280 (una immagine del sensore è

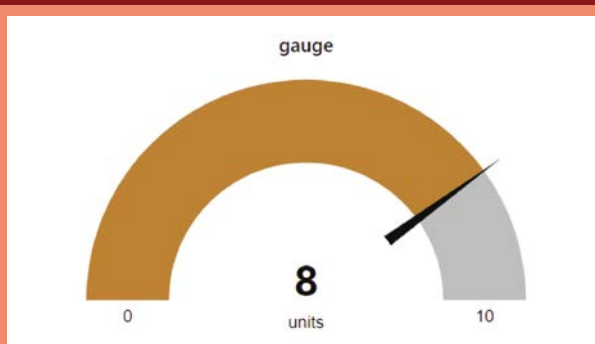


Fig. 10 - Dashboard ottenuta dall'esempio.

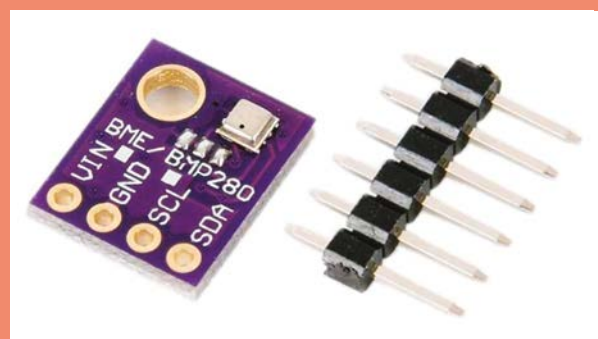


Fig. 11 - Sensore BME280.

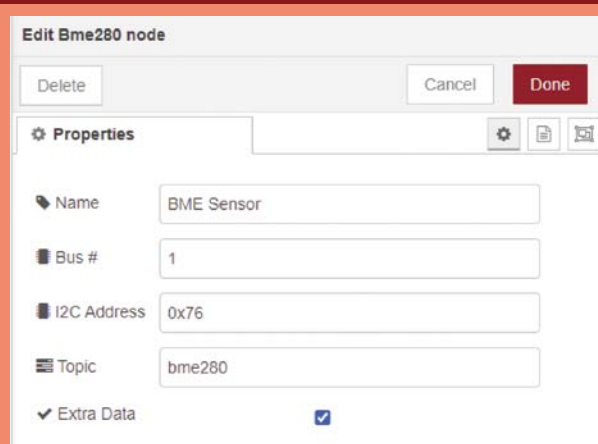


Fig. 12 - Configurazione nodo BME280.

riportata in Fig. 11).

La realizzazione è semplicissima: partiamo installando il nodo specifico che ci permette di comunicare con il sensore. Come di consueto usiamo la pagina di ricerca (manage palette, accessibile dal menu con il simbolo delle tre linee orizzontali in alto a destra), cerchiamo il nodo *node-red-contrib-bme280* ed installiamolo.

Una volta installato il nodo, posizioniamolo sul workspace ed apriamo la sua pagina delle proprietà. Dobbiamo:

- Rinominarlo in BME Sensor
- Assegnare come bus 1
- Assegnare come I2C Address 0x76
- Assegnare come topic bme280
- Spuntare l'opzione extra data (ci permetterà di leggere anche la temperatura apparente) come illustrato in Fig. 12.

Il nodo ha un funzionamento semplicissimo, ogni volta che riceve un messaggio in ingresso, ritorna un object con 4 campi:

- Temperatura (temperature_C)
- Umidità (humidity)
- Pressione barometrica (pressure_hPa)

- Temperatura apparente (heatIndex)

Aggiungiamo adesso gli altri nodi necessari.
Ci occorre:

- un nodo inject per triggerare la lettura periodicamente (decidiamo di leggere 1 volta al secondo)
- 4 nodi change (ci serviranno per estrarre i campi dal messaggio in output dal nodo BME280)
- 3 nodi function (ci serviranno per formattare l'output di umidità, pressione e temperatura apparente, che hanno molte cifre decimali)
- 4 nodi Gauges (per la visualizzazione)

Creiamo anche l'infrastruttura della dashboard, composta da un Tab e da un Group, ed associamo i 4 gauges allo stesso gruppo, ed eseguiamo le seguenti configurazioni.

- Gauge 1:
 - Nome: Temperature
 - Range: -20 – 80
 - Units: °C
- Gauge 2:
 - Nome: Humidity
 - Range: 0 – 100
 - Units: %
- Gauge 3:
 - Nome: Pressure

Fig. 13 - Configurazione del gauge Temperature.

Fig. 14 - Configurazione del nodo Change che estrae il dato di temperatura.

- Range: 500 – 1500
- Units: hPa
- Gauge 4:
 - Nome: Heat Index
 - Range: -20 – 80
 - Units: °C

In Fig.13 è rappresentata la configurazione del gauge che fornisce l'indicazione relativa alla temperatura.

A questo punto configuriamo i nodi change. Lo scopo di questi nodi è quello di estrarre un campo specifico dal messaggio contenente l'object complessivo e passarlo al nodo function che lo formatterà correttamente per la visualizzazione (o direttamente al nodo gauge nel caso del dato di temperatura).

Per farlo ci serviamo della regola set e impostiamo nel campo "to" il valore `msg.payload.<campo specifico>`. Dove il valore `<campo specifico>` individua il campo che vogliamo estrarre dal messaggio. In Fig.14 è rappresentata la configurazione del nodo change che estrae il dato di temperatura.

A questo punto non ci rimane che configurare i 3 nodi function. Questi nodi hanno la funzione di formattare cor-

Fig. 15 - Uno dei nodi function usati per correggere la precisione.

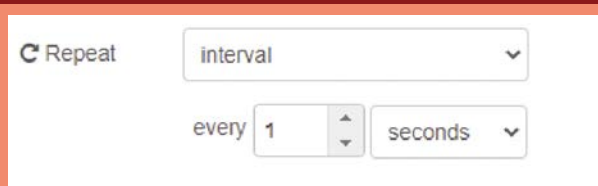


Fig. 16 - Configurazione del nodo inject.

rettamente i dati di alcuni campi (umidità, pressione, heat index), che hanno una precisione eccessiva (svariate cifre decimali) per la tipologia di dato da visualizzare. Tramite i nodi function, servendoci del metodo *toFixed()* portiamo la precisione a 2 cifre decimali. Lo snippet di codice Javascript che esegue questa operazione è riportato in Fig.15 ed è lo stesso per tutti i 3 i dati da manipolare (la temperatura viene già riportata dal nodo BME280 con una precisione di 2 cifre decimali). Ora non ci resta che configurare anche il nodo inject per fornire la periodicità. Come abbiamo visto anche nelle puntate precedenti, è sufficiente settare l'opzione interval ed indicare 1 secondo come periodicità (Fig.16). A questo punto la configurazione è conclusa, e possiamo collegare i nodi del nostro flow come riportato in Fig.17. Ora possiamo eseguire il deploy del nostro flow ed effettuare i collegamenti tra la nostra Raspberry Pi ed il sensore. La breakout BME280 ha 4 pin: Alimentazione,

GND, SDA ed SCL. Colleghiamo il pin di alimentazione ad una delle uscite a 5V della Raspberry Pi e il GND ad uno dei GND. I pin SDA ed SCL invece vanno collegati ai rispettivi pin della porta I2C1 della Raspberry Pi (pin 3 e 5 del connettore GPIO). A questo punto, se tutto è stato eseguito correttamente, collegandoci alla pagina:

<Node-RED server URL/IP>:1880/ui

Dove <Node-RED server URL/IP> rappresenta l'indirizzo della nostra raspberry, dovremmo visualizzare la dashboard rappresentata in Fig.18.

CONCLUSIONI

Siamo arrivati alla conclusione di questa quarta puntata in cui abbiamo visto come gestire la security e il componente dashboard in Node-RED. Le opzioni di security ci permettono di proteggere adeguatamente il nostro editor e le nostre interfacce grafiche qualora decidessimo di esporle sulla rete, mentre la dashboard ci permette di creare in maniera estremamente semplice interfacce grafiche hostate dal server, che quindi possono essere raggiungibili tramite diversi dispositivi, come PC, tablet e smartphone, ma è anche possibile visualizzarle su un dispositivo di output locale, come un monitor o un display, e creare quindi interfacce uomo-macchina molto evolute. □

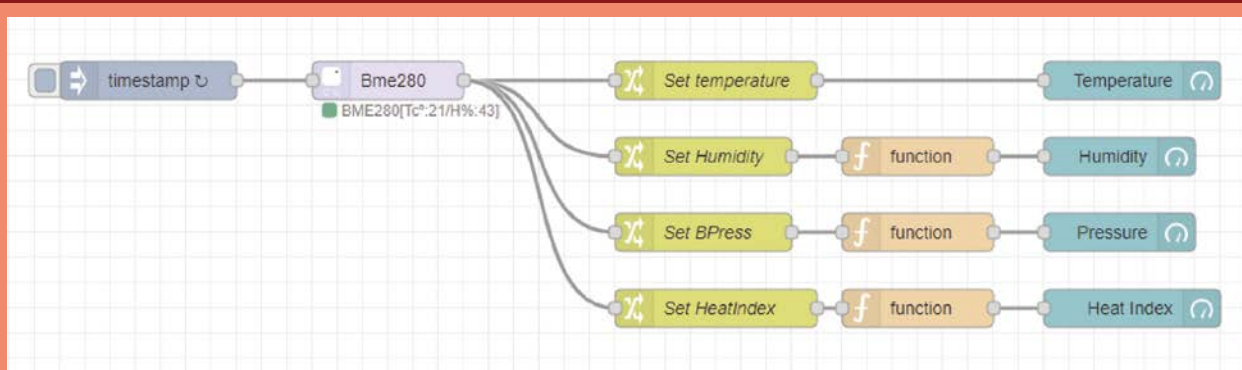


Fig. 17 - Flow con i collegamenti completi.



Fig. 18 - Dashboard della stazione meteo.