

Il mondo dell'



dell'Ing. MIRCO SEGATELLO



Dopo aver introdotto l'argomento IoT, vediamo come si può utilizzare l'infrastruttura di rete WiFi per far comunicare dispositivi domestici.

Continuiamo questo corso inerente ai dispositivi connessi, vale a dire all'Internet delle cose, partendo da dove eravamo rimasti alla fine della prima lezione; in essa abbiamo affrontato l'argomento dal punto di vista generale, più che altro per capire cosa si intende con IOT e quali siano le potenzialità di questa tecnologia.

Ora invece entriamo più in dettaglio e proviamo a costruire il nostro primo dispositivo connesso. Requisito per tale progetto è la disponibilità di una connessione che permetta al nostro dispositivo di accedere alla rete Internet; la soluzione decisamente più pratica ed economica è accedere alla nostra

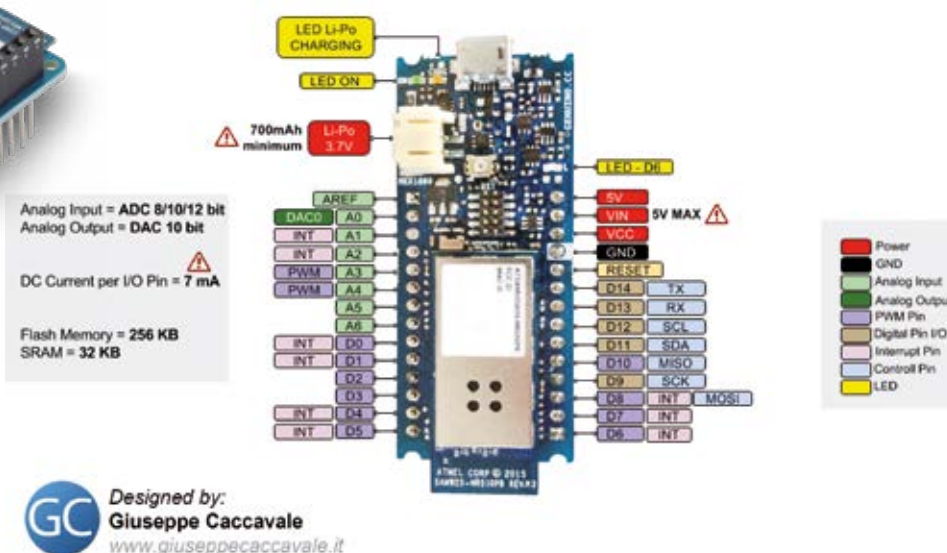
rete domestica a sua volta interfacciata con il mondo di internet. Il secondo requisito è disporre di un "oggetto" che possa connettersi alla rete come ad esempio una scheda Arduino dotata di un Ethernet Shield. In questa puntata utilizzeremo, tuttavia, una scheda di più recente produzione (presentata a maggio 2018) e facente parte del mondo Arduino: si tratta della board MKR1000 visibile nella **Fig. 1** e disponibile sul sito www.futurashop.it con il codice MKR1000.

Si tratta di una scheda di prototipazione in formato Arduino MKR, basata sul microcontrollore (MCU) con architettura a 32 bit Cortex M0, alla quale è stato aggiunto un modulo WiFi.



Fig. 1 - Scheda MKR1000 e relativa pin-out.

MKR1000 PINOUT



Tutte le schede MKR sono compatibili con l'IDE classico di Arduino che permette la solita immediatezza e facilità di utilizzo; l'unica attenzione richiesta è che la serie MKR funziona a 3,3 volt e non è tollerante nei confronti dei segnali a 5V, quindi se si interfaccia la board con una scheda che fornisce segnali TTL standard, ossia 0/5V, si rischia il danneggiamento del microcontrollore. Per poter utilizzare questa scheda è necessario impostare l'ambiente di programmazione aggiornando la compatibilità alle schede con microcontrollore Atmel

SAMD; per fare ciò, apriamo l'IDE di Arduino, da qui passiamo nel "Gestore schede" e cerchiamo la libreria di nome "Arduino SAMD Boards (32-bit Arms Cortex MO+) by Arduino". Completata l'installazione (**Fig. 2**) vedremo nell'elenco delle schede supportate, comparire anche tutte le schede MKR.

IL SETUP SUL COMPUTER

Connettete ora la scheda al Personal Computer: Windows (stiamo ipotizzando di lavorare su un sistema Windows...)

Fig. 2 - Installazione libreria per compatibilità con MCU Cortex MO.

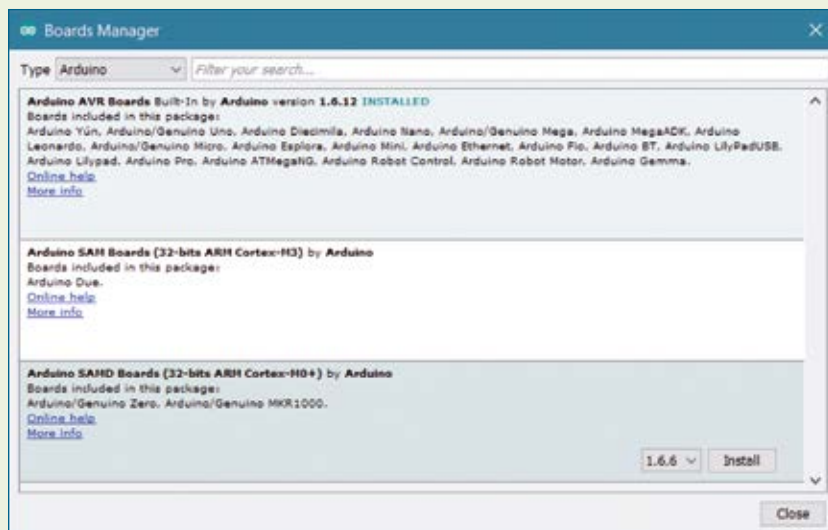


Fig. 3

Installazione driver per scheda MKR1000.





Fig. 4 - Struttura del sistema IoT.

vi chiederà l'installazione dei driver; specificate che volete l'installazione dei driver in locale e cercate la cartella "Drivers" all'interno dei file di Arduino.

Se ci dovessero essere degli intoppi accedete alla "Gestione dispositivi" di Windows e cliccate con il pulsante destro del mouse sopra la periferica non riconosciuta, quindi avviate "Aggiorna driver". Il classico esempio Blink che trovate sempre tra quelli dell'IDE Arduino andrà più che bene per testare il funzionamento della scheda.

Le schede MKR hanno la peculiarità di non avere un chip dedicato alla comunicazione seriale (come avviene con Arduino UNO) in quanto il microcontrollore dispone già della periferica integrata; sfortunatamente la porta di comunicazione virtuale potrebbe -in alcuni casi- creare problemi nella programmazione o nell'uso del Serial Monitor di Arduino. Per ripristinare

eventuali "inceppamenti" potete procedere con il classico reset, ossia premendo l'apposito pulsante sulla scheda che provvede al reset del microcontrollore e della comunicazione USB; se fate ciò, Serial Monitor deve essere chiuso e poi riaperto. Adesso siamo pronti ad entrare nel vivo dell'applicazione, prendiamo in considerazione una semplice applicazione che prevede di leggere la temperatura e l'umidità presenti in un locale e di attivare un riscaldamento all'occorrenza, ovviamente con la possibilità di controllare il tutto da remoto tramite il nostro smartphone, in puro stile IoT.

Per attivare questa funzione è necessario appoggiarsi ad un servizio cloud online che permetta di far comunicare il nostro smartphone con il dispositivo remoto tramite un'opportuna interfaccia (**Fig. 4**).

Tra i tanti servizi disponibili abbiamo scelto il cloud Cayenne che permette, come vedremo, di ottenere delle interfacce molto accattivanti in modo semplice e completamente gratuito. Iniziamo subito accedendo al sito di riferimento <https://mydevices.com> e creiamo un account cliccando in alto a destra sulla voce "SIGN UP FREE".

Il secondo passo è scaricare la libreria dedicata, pertanto apriamo il gestore di librerie di Arduino, cerchiamo ed installiamo quella di nome CayenneMQTT (**Fig. 5**); la libreria mette a disposizione diversi esempi che coprono praticamente qualsiasi esigenza e compatibilità con tutte le principali schede in commercio.

APPLICAZIONE PRATICA CONTROLLO DI PARAMETRI AMBIENTALI

Per farvi vedere le potenzialità di questo sistema prendiamo in considerazione un esempio concreto, che riguarda il controllo da remoto dei parametri ambientali riguardanti una stanza della vostra casa; qui vogliamo poter controllare da remoto un riscaldamento per climatizzare l'ambiente secondo le necessità che dovessero sorgere.

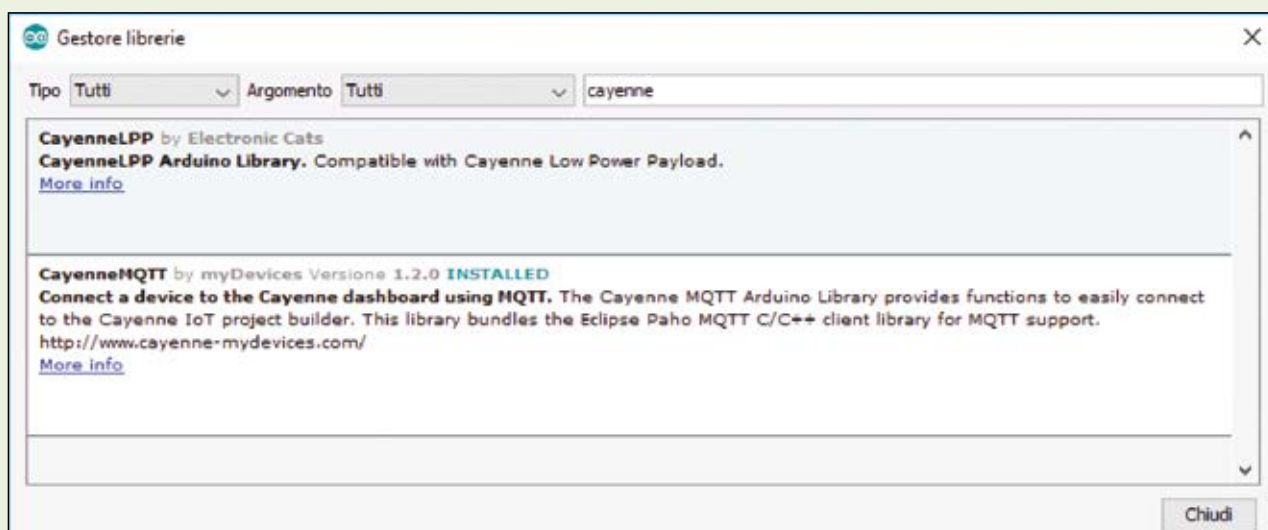


Fig. 5 Installazione libreria CayenneMQTTcomponenti.

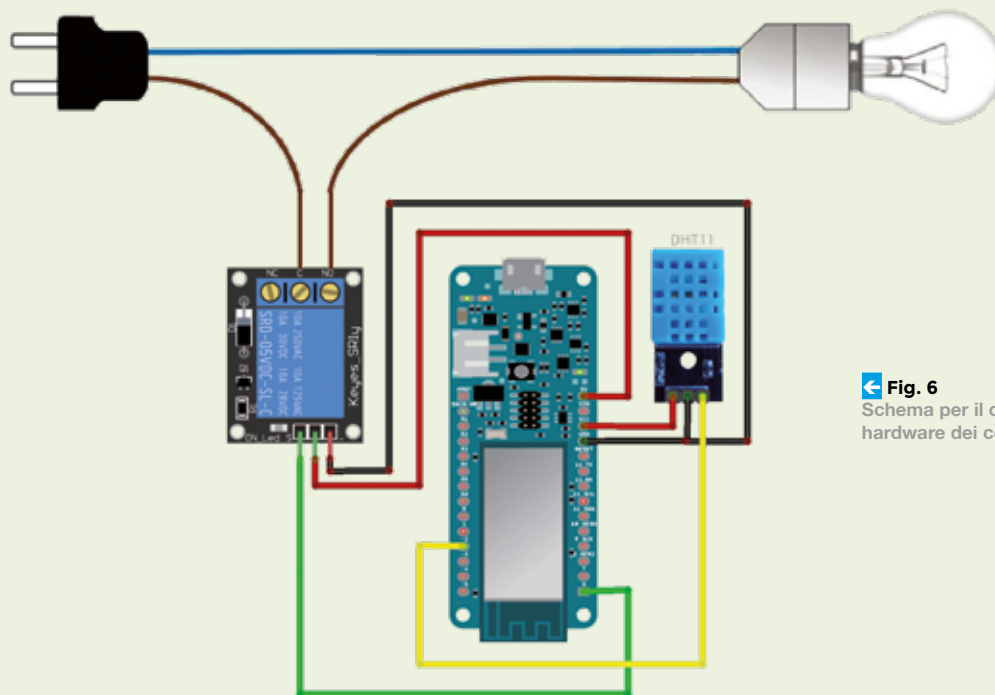


Fig. 6
Schema per il collegamento hardware dei componenti.

Il materiale necessario oltre ovviamente alla scheda Arduino MKR1000, sarà un sensore DHT11 per la misura di temperatura e umidità ed un modulo relé per l'attivazione di una stufetta elettrica idonea al riscaldamento della stanza. La scheda MKR1000 potrà essere alimentata con un semplice alimentatore a presa integrata (wall-cube) avente tensione di uscita di 5V e cavetto terminante con uno spinotto microUSB, come quello usato per ricaricare gli smartphone. Il tutto, secondo lo schema mostrato nella **Fig. 6**.

Andiamo ora sulla schermata principale del cloud Cayenne, dove, essendo il primo accesso, verremmo guidati nella procedura di configurazione in soli tre semplici passi. Il primo passo consisterà nel selezionare la scheda utilizzata: nel nostro caso è Arduino.

Nel secondo passo sono visualizzate le istruzioni per rendere operativo il sistema, ovvero connettere la scheda Arduino al

PC e avviare l'IDE per la programmazione (**Fig. 7**).

Nel terzo e ultimo passo dobbiamo selezionare il tipo di scheda utilizzata, che nel nostro caso è una Arduino MKR1000 (vedere **Fig. 8**).

Appena selezionata la scheda si aprirà un pop-up con il listato già pronto per essere caricato nella scheda, si tratta di un semplice esempio di invio di un dato (il conteggio dei millisecondi) dalla scheda al cloud.

Copiate il listato nell'IDE di Arduino ed inserite le credenziali di accesso alla vostra rete WiFi (ssid e wifiPassword) quindi programmate Arduino ed attendete che Cayenne riconosca la connessione della scheda, al termine si aprirà una schermata che mostrerà in tempo reale i millisecondi da quando è stata programmata la scheda e contemporaneamente Cayenne invierà sul vostro account una mail di conferma.

Nella **Fig. 9** è proposta la schermata con l'esempio predefi-

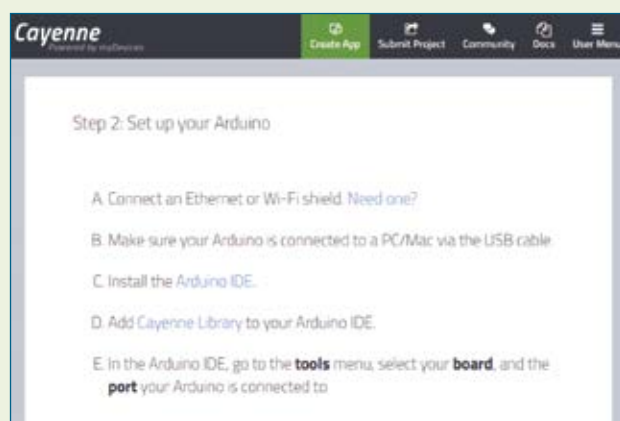


Fig. 7 - Le istruzioni fornite da Cayenne.

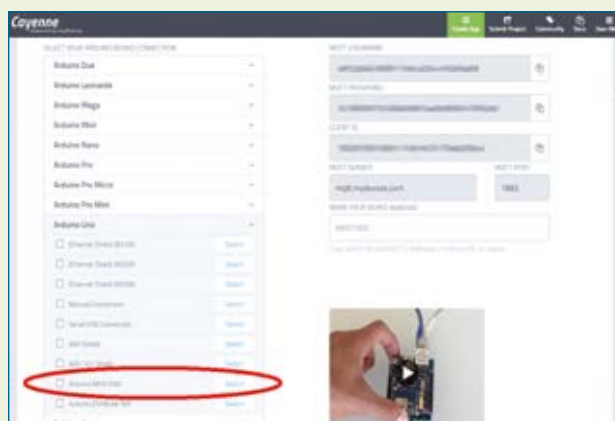


Fig. 8 - Selezione tipo di scheda e copia dello sketch.

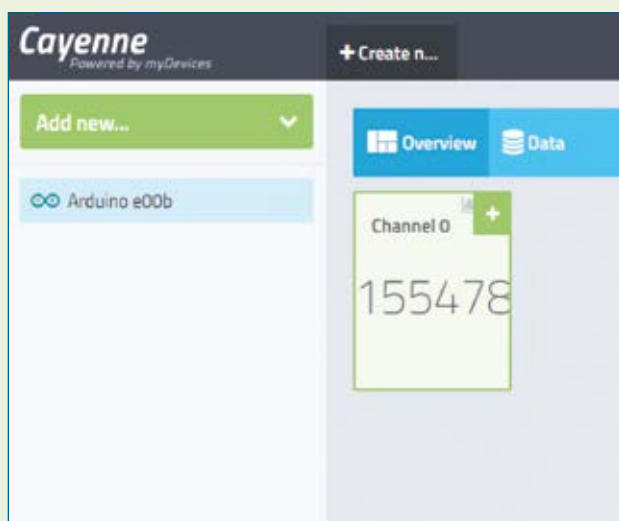


Fig. 9 - Esempio predefinito in esecuzione

nito in esecuzione. Cliccando sul pulsante a forma di ingranaggio presente sulla destra potete modificare il nome del dispositivo ed assegnargli un'icona. Nel listato noterete i campi MQTT Username, MQTT Password e Client ID che rappresentano le credenziali di accesso al cloud abbinate al vostro dispositivo. Adesso siamo pronti per personalizzare la nostra applicazione eseguendo per prima cosa i collegamenti elettrici del sensore DHT11 e del relé (riferirsi alla **Fig. 6**). Aprite lo sketch di nome MKR1000_cayenne_MQTT_DHT11.ino fornito assieme ai file della rivista, modificate i campi relativi all'accesso alla vostra rete WiFi e quelli relativi all'accesso al cloud, e caricatelo sulla scheda. In sintesi, questo sketch legge ad intervalli regolari il valore di temperatura e di umidità dal sensore DHT11 e li invia ad intervalli regolari a Cayenne per la visualizzazione, contemporaneamente rimane in ascolto dell'invio del comando per attivare l'uscita alla quale è connesso il relé che comanda la stufetta di riscaldamento. Per motivi di sicurezza viene anche monitorato il pin di Arduino che comanda il relé ed ogni volta che vi è una variazione di stato ne viene dato avviso al cloud, in questo modo avrete conferma di ricezione del comando di spegnimento ed accensione. Comprendere il meccanismo di funzionamento di Cayenne è molto semplice e tutto si riduce alla riga di programma:

Cayenne.virtualWrite(CHANNEL_ID, VALUE)

che permette l'invio del dato di valore VALUE al canale di nome CHANNEL_ID. Cayenne archivia i dati ricevuti ed è possibile creare una dashboard personalizzata per visualizzare questi dati nel modo che si ritiene più opportuno e la forza di questo cloud sta proprio nella facilità di gestione di questa funzione. E' infatti sufficiente cliccare su "Add new...", selezionare "Device/Widget" e successivamente "Custom Widgets" ed inserire due campi numerici di nome "Value", due grafici di

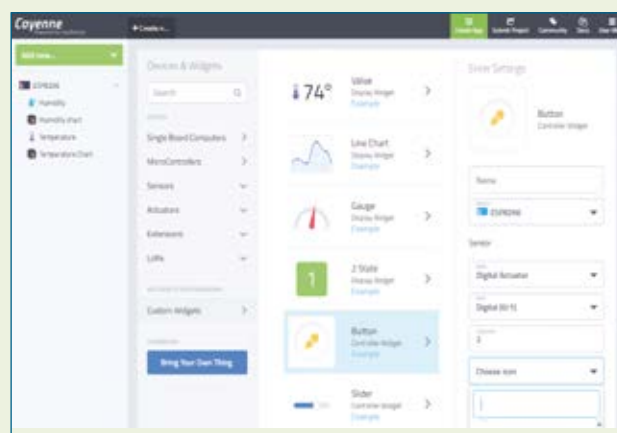


Fig. 10 - Inserimento dei widget.

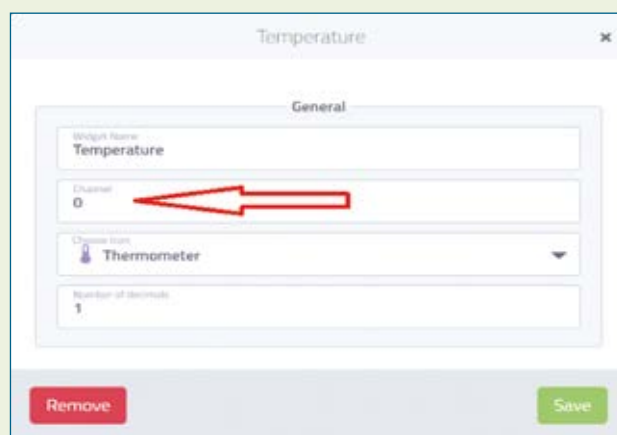


Fig. 11 - Modifica dei parametri del widget per la temperatura.

nome "Line chart" un pulsante di nome "Button" e un campo numerico di nome "Value" (**Fig. 10**).

Per ciascun widget potete personalizzare il nome e l'icona ma la cosa importante è specificare a quale canale sia associato; nel nostro esempio il canale 0 è la temperatura, il canale 1 è l'umidità, il canale 2 è il pulsante che comanda il relé ed il canale 3 è lo stato del pin che comanda il relé.

Anche dopo aver inserito i widget, cliccando sui pulsantini a forma di ingranaggio potete modificare i parametri a piacere attraverso la schermata visibile in **Fig. 11**.

Alla fine vi ritroverete una schermata come quella visibile in **Fig. 12**. Aprendo Serial Monitor di Arduino avrete anche a disposizione un debug delle operazioni svolte, utile per verificare che tutto funzioni correttamente (**Fig. 13**).

Da qualsiasi dispositivo connesso alla rete Internet potrete accedere alla vostra dashboard, visualizzare le condizioni climatiche della stanza e decidere, in base ai valori letti, se attivare o meno il riscaldamento.

Ovviamente le possibili funzioni non si limitano a questo, ma è anche possibile attivare degli allarmi al verificarsi di determinate situazioni. Anche in questo caso Cayenne mette a disposizione una procedura davvero semplicissima, basata sul

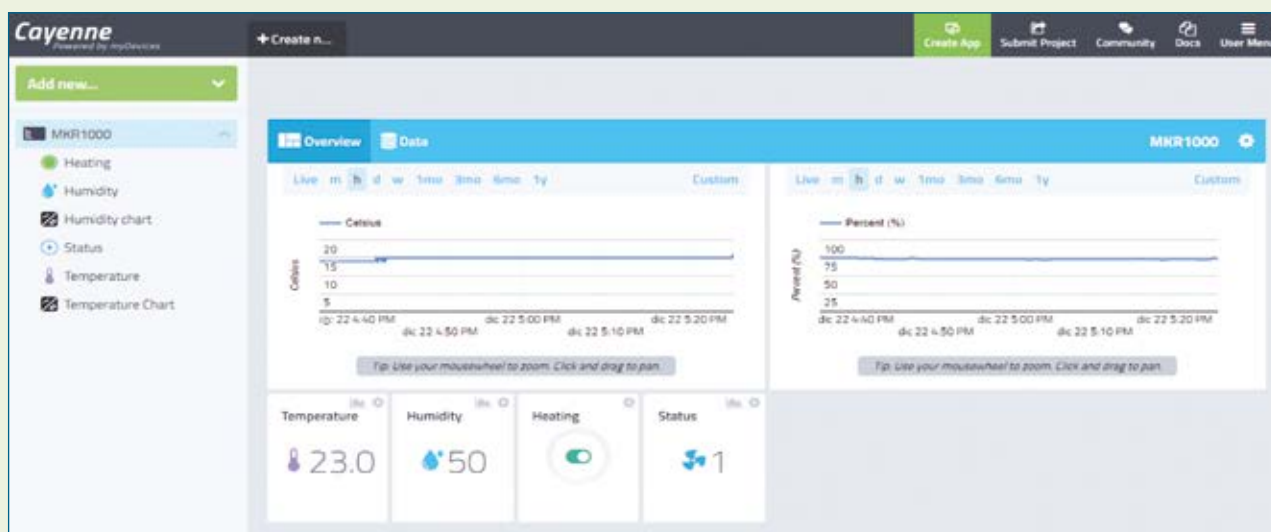


Fig. 12 - Dashboard configurata per l'esempio.

principio IF-THEN come visibile nella **Fig. 14**, la quale propone, nello specifico, la configurazione di un evento di trigger. È sufficiente cliccare su "Add new..." e selezionare "Trigger", quindi nella schermata che si aprirà diverrà possibile definire quale canale genererà l'evento e per quale valore; nel nostro caso andremo a monitorare la temperatura e generare un avviso se questa dovesse scendere sotto i 10 gradi. All'evento è possibile associare (**then**) l'invio di una e-mail (tale funzionalità richiede che venga specificato l'indirizzo di posta elettronica destinatario) oppure di un messaggio di testo (in questo caso il servizio richiede la definizione di un numero di telefono destinatario). Ma non finisce qua: esiste anche la funzione "Event" con la

quale è possibile generare un evento in un certo giorno ad un'ora specificata; nel nostro caso potremmo, ad esempio, attivare (o disattivare) il riscaldamento in un ben preciso momento della giornata e ovviamente potremmo configurare la relativa notifica che verrà inviata tramite i canali consueti (**Fig. 15**).

ANCHE DA APP

Non poteva mancare l'app Cayenne, disponibile gratuitamente sul Play Store della Google, la quale permette di fare tutte le operazioni essenziali per gestire il nostro dispositivo, compresa la gestione dei trigger. Anche l'app è votata alla semplicità ed è molto intuitiva da usare, pertanto non ci dilungheremo sulla sua descrizione. Un'ultima nota, prima di concludere, riguarda la modalità di alimentazione della scheda MKR1000: i dati ambientali sono inviati ad intervalli regolari ma da remoto possiamo inviare il comando di attivazione dell'uscita in qualsiasi momento, per cui la scheda sarà sempre connessa al WiFi e non sarà possibile abilitare una qualche modalità di risparmio energetico. Insomma, il modulo wireless dovrà restare costantemente alimentato, con ciò che ne consegue in termini di consumo energetico, il che, nelle applicazioni in mobilità, dev'essere ben valutato. In ogni caso la scheda MKR non dispone di una vera e propria modalità di deep sleep che le permetta di assorbire solo pochi microampere, come è per altri dispositivi, pertanto un'eventuale alimentazione a batteria (prevista nella scheda) permetterebbe il funzionamento solo per qualche giorno e non certamente per anni. Se la vostra applicazione richiede una durata elevata, dovrete prevedere una sorgente di alimentazione esterna come ad esempio un piccolo pannello solare.

CONCLUSIONI

Termina qui questa seconda puntata del nostro corso, dove siamo passati dalla teoria alla pratica, proponendovi un pro-

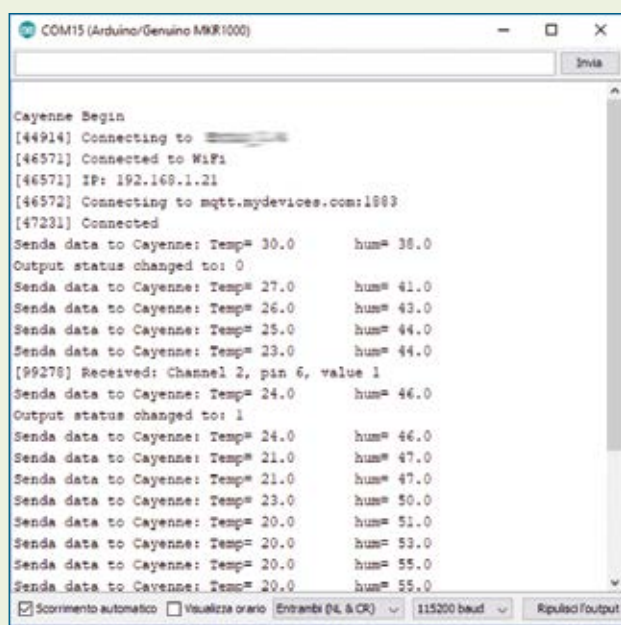


Fig. 13 - Schermata di Serial Monitor durante l'esecuzione dello sketch.

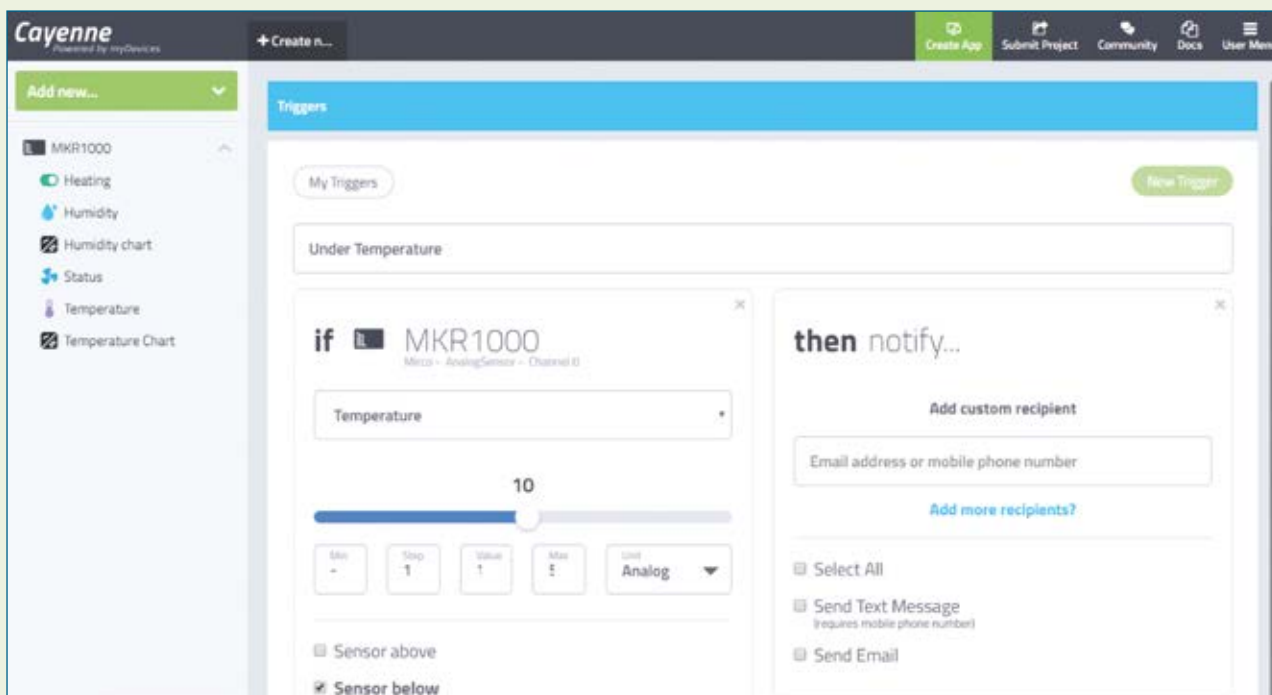


Fig. 14 - Configurazione di un evento di trigger.

getto finalizzato al controllo remoto, tramite accesso WiFi ad Internet, dei parametri ambientali; un controllo bidirezionale, che ci permette sia di monitorare l'ambiente in cui il sistema

si trova, sia di comandare un riscaldatore o altro genere di attuttore all'occorrenza. Vi diamo appuntamento a quella successiva in cui vedremo altri modi di fare IOT.

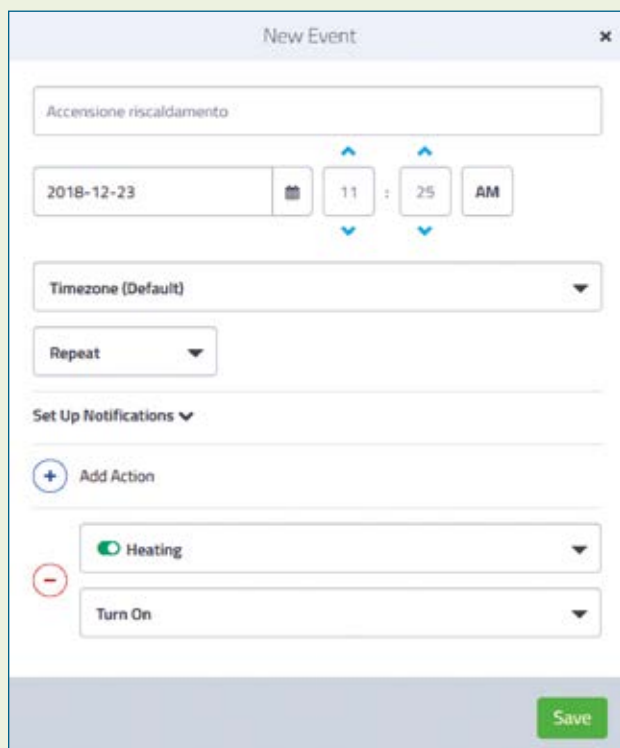


Fig. 15 - Generazione di un evento.

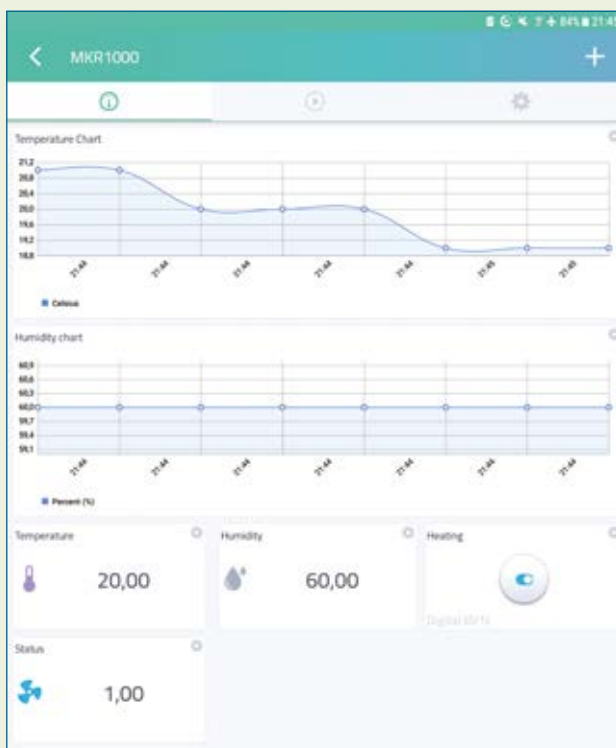


Fig. 16 - Schermata dell'app Cayenne con la nostra dashboard.