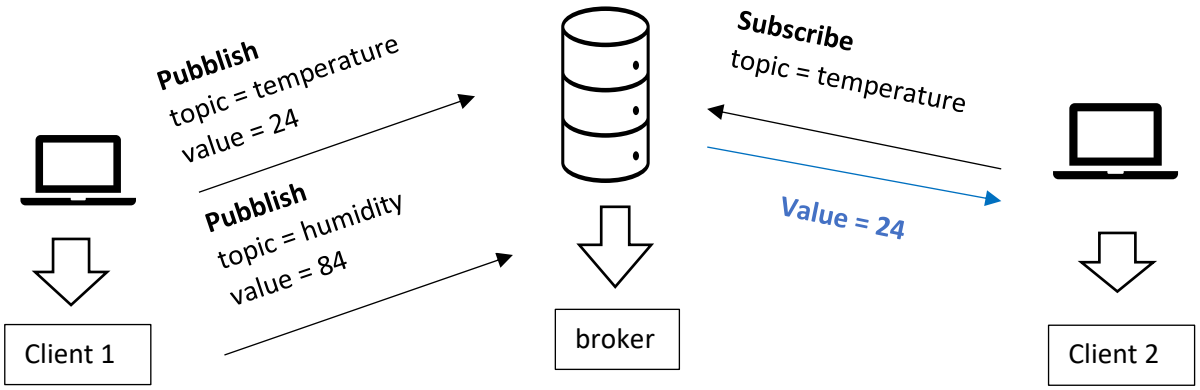
Iot Bridge Platform

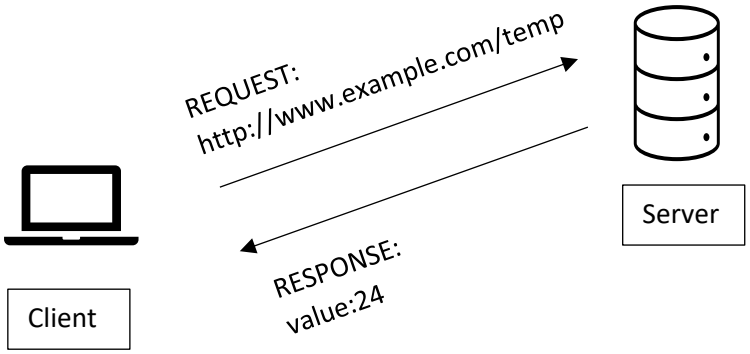
Autore: Andrea Baroni – Matricola: 0000855821

Introduzione

Lo scopo del progetto è la realizzazione di un tool per la **visualizzazione**, il **salvataggio** e la **conversione** di dati acquisiti tramite i seguenti protocolli: MQTT, COAP e HTTP. Sebbene i metodi di acquisizione dei dati citati precedentemente differiscano, è possibile raggrupparli in due macrocategorie denominate **publish/subscibe** e **request/response**. La categoria pubblish/subcribe di cui fa parte il protocollo MQTT può essere schematizzata tramite la seguente figura:



Un client può fungere sia da pubblisher che da subscriber e non ha alcuna conoscenza di chi riceverà il messaggio che pubblica o da quale client riceverà il messaggio a cui si è sottoscritto. Inoltre, è sempre presente un broker che si occupa di smistare i messaggi ricevuti in base al topic. Un' ulteriore considerazione è che sul piano temporale la fase di pubblish e quella di subscribe sono completamente separate, un client che si sottoscrive ad un topic non saprà se e quando riceverà il messaggio desiderato. Il paradigma request/response, schematizzato nella figura sottostante, di cui fanno parte i protocolli COAP e HTTP, al contrario non presuppone la presenza di un broker, e tipicamente i ruoli sono divisi tra server e client; le risorse sono indicate tramite URL e non attraverso topic, inoltre il server deve soddisfare ogni richiesta singolarmente.



Uno degli scopi del progetto è la conversione di dati ricevuti da un protocollo di tipo pubblish/subscribe tipo MQTT verso un protocollo di tipo requst/response come COAP e HTTP, in particolare come verrà dettagliato nella prossima sezione avendo a disposizione un canale in input da uno dei tre protocolli citati

precedentemente verrà creato un canale in output verso un protocollo diverso da quello di acquisizione a cui alti client potranno connettersi. Per quanto riguarda il salvataggio si è utilizzato Influx 2.0 un time series database il quale oltre al salvataggio permette anche la creazione di alert personalizzati. In tutte e tre le fasi visualizzazione, salvataggio e conversione è presente la funzionalità di filtraggio che permette di selezionare tramite soglie di minimo e massimo quali dati debbano essere processati. Sebbene Influx 2.0 abbia un'interfaccia grafica che permetta di impostare tutti i necessari valori per il salvataggio e gli alert, sono stati predisposti script per la gestione, tramite le API (1), delle fasi di predisposizione del database e creazione delle soglie di allerta.

Progettazione

In questa sezione si descrivono le funzionalità ad alto livello del tool, i dettagli implementativi vengono mostrati nella prossima sezione. In tutte e tre le funzionalità descritte è possibile specificare delle soglie di minimo e di massimo che permettono di filtrare i dati in ingresso prima di visualizzarli, salvarli o trasferirli verso alti protocolli.

Visualizzazione

Per quanto riguarda il protocollo MQTT la visualizzazione consiste nello stampare a video i valori ricevuti dal broker per un particolare topic ogni volta che vengono ricevuti. Per i protocolli http e coap invece si esegue la richiesta, si aspetta di ricevere una risposta, dopo aver ottenuto la risposta si attende un numero prefissato di secondi e si reitera la richiesta. È importante sottolineare che per quanto riguarda i protocolli coap e http solo dopo aver ottenuto il risultato di una richiesta si passa ad eseguire la richiesta successiva.

Salvataggio

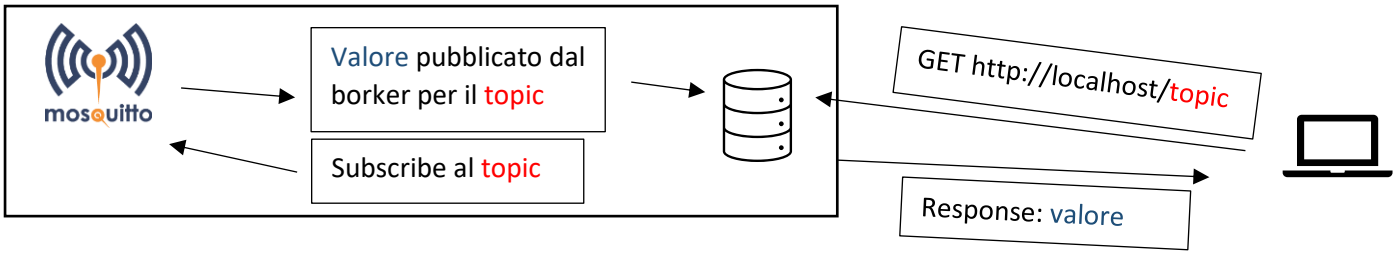
La parte di salvataggio consiste nel salvare in un database INFLUX 2.0 i dati ricevuti dai protocolli, essenzialmente si riutilizza il codice impiegato per la visualizzazione e al posto di visualizzare a video i risultati ottenuti, li si salva sotto forma di coppie <tempo, valore> nel database INLFLUX. Le fasi di settaggio del database come la scelta del nome del bucket, la politica di retention ed eventuali alert nel caso si superino delle soglie prefissate ed anche la scelta dei canali sui cui mostrare gli alert possono essere impostati tramite l'interfaccia di influx ma per rendere il tool auto contenuto si sono predisposti script che eseguono le operazioni appena citate.

Conversione

La conversione consiste nel trasformare un flusso di dati in ingresso in un particolare protocollo in un flusso di dati in uscita in un altro protocollo diverso da quello in input. Avendo considerato tre protocolli mqtt, coap e http i casi da considerare sono sei:

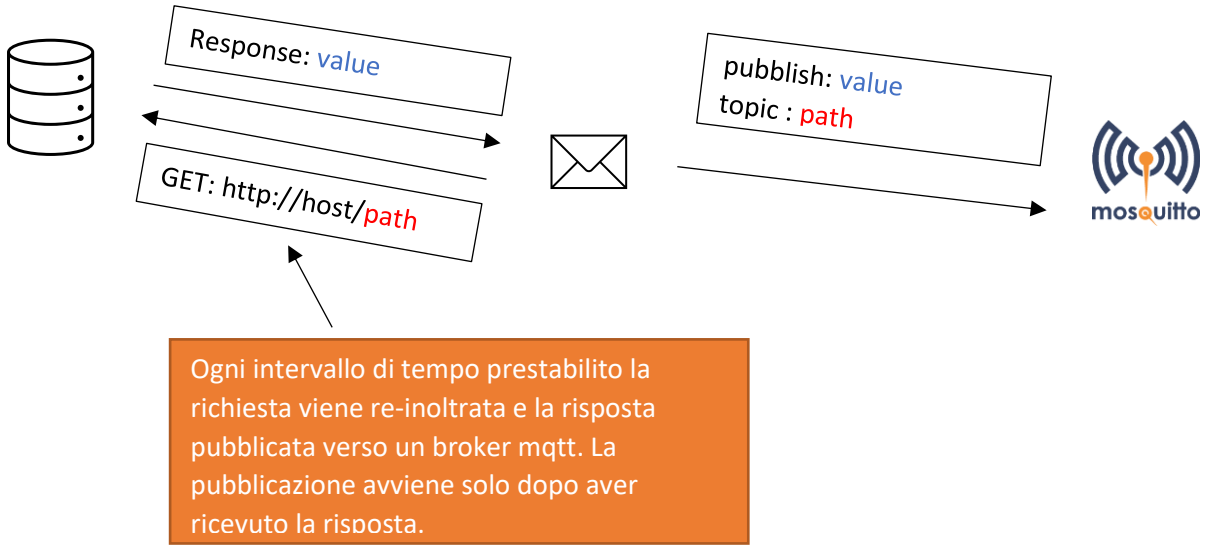
Flusso dati in ingresso	**Parametri da specificare per il flusso dati in ingresso**	**Flusso dati in uscita**
mqtt	Topic, porta, indirizzo del broker	coap
mqtt	Topic, porta, indirizzo del broker	http
coap	URL, porta	mqtt
coap	URL, porta	http
http	URL, porta	mqtt
http	URL, porta	coap

La conversione da mqtt a coap e da mqtt a http viene gestita nel seguente modo: come viene illustrato nella figura sottostante un flusso di esecuzione rimane in ascolto di ricevere un messaggio di risposta per un particolare topic, la risposta viene salvata in una variabile e nel momento in cui un client si connette al



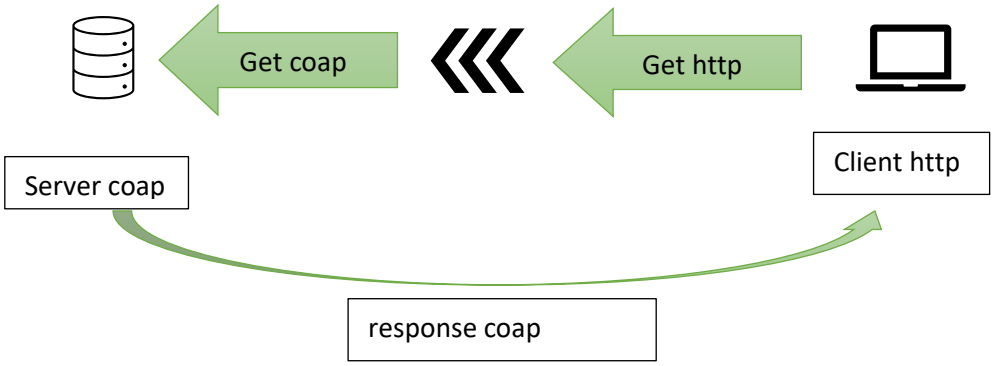
server coap o http in cui avviene la conversione con il path uguale al topic, viene restituito il risultato restituito in precedenza dal broker.

La conversione da coap a mqtt e da http a mqtt è descritta dalla figura sottostante. All'avvio viene eseguita una richiesta al server coap o http, che ogni intervallo di tempo prestabilito viene rieseguita solo dopo aver ricevuto la risposta.



La risposta ottenuta dal server coap o http viene pubblicata verso un broker mosquitto avente il topic uguale al path della richiesta.

La conversione da http a coap o da coap a http può essere illustrata dalla seguente figura in cui si considera la conversione da coap a http:



Ogni richiesta verso un server http verrà inoltrata verso un server coap che invierà l'eventuale risposta verso il client http. Nella figura sopra, nel caso in cui il client http non ricevesse una risposta entro un tempo stabilito andrebbe in timeout, per cui si è dovuto gestire questo caso.

Implementazione

Il tool è stato realizzato tramite il linguaggio Javascript/node ed ha un'interfaccia a linea di comando. L'intero codice sorgente è disponibile alla pagina Github seguente (1) nel cui readme viene indicato l'utilizzo dei principali programmi. Per quanto riguarda l'organizzazione generale del codice si è provveduto a creare tre moduli corrispondenti alle tre funzionalità descritte precedentemente. Il file `display.js` si occupa di recuperare i dati prodotti dai protocolli e a visualizzarli a video, inoltre mette a disposizione delle api per il recupero dei dati dai protocolli in modo da rendere il modulo riutilizzabile. Per quanto riguarda la visualizzazione dei valori dai protocolli coap e http, tramite la funzionalità degli eventi messa a disposizione da Javascript/node non appena si riceve la risposta dalle api per il recupero, si stampa il risultato a video, si aspetta un numero prefissato di secondi e si "emette" un evento che fa inoltrare una nuova richiesta al server. Per implementare la funzionalità che recupera i dati dal protocollo mqtt, si è provveduto a restituire, tramite una callback, uno stream in sola lettura che conterrà man mano che i dati verranno prodotti i valori pubblicati dal broker. Il file `save.js` è responsabile del salvataggio dei dati in un database Influx 2.0. Le informazioni richieste per salvare i dati all'interno del database, specificate da un file json sono:

1. **L'url** a cui si trova il database,
2. **un token** per permettere l'accesso al database
3. **il nome** dell'organizzazione a cui appartiene il database
4. il tag **measurement** che sarebbe l'analogo del nome delle tabelle in un database relazionale
5. un capo obbligatorio chiamato **field**
6. eventuali tag aggiuntivi definiti dal'utente

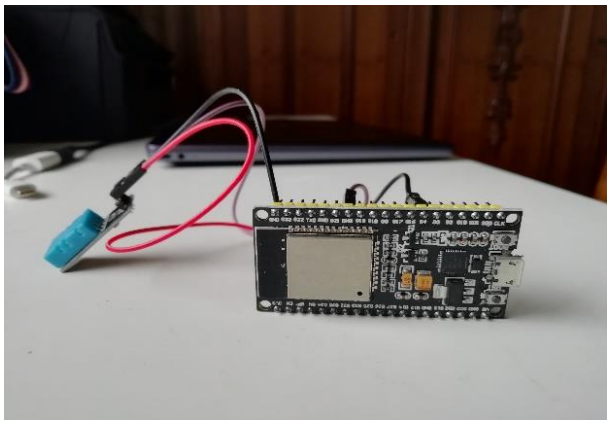
Al fine di automatizzare il settaggio del database Influx 2.0 a corredo del tool, si sono implementati alcuni script per le seguenti operazioni:

1. Creazione di un bucket con annessa politica di retention,
2. Creazione di check per controllare che i valori inseriti all'interno del database non superino una soglia prefissata, e nel caso settare un'apposita variabile chiamata status, che può assumere valori come WARN, INFO, OK, CRIT. Il linguaggio mediante il quale vengono specificati i check prende il nome di Flux ed è descritto al seguente link (2)
3. Creazione di endpoint per permettere di avvisare l'utente nel caso succedano eventi specificati tramite rules anche esse definite dall'utente. Tipici endpoint sono invio di un messaggio verso un canale telegram o l'esecuzione di una post ad un particolare URL.

Per eseguire le operazioni nel database influx si sono utilizzate le api descritte al seguente link (3). La parte di conversione tra i protocolli è gestita dal file `bridge.js` e realizza passo a passo i passaggi descritti nella precedente sezione.

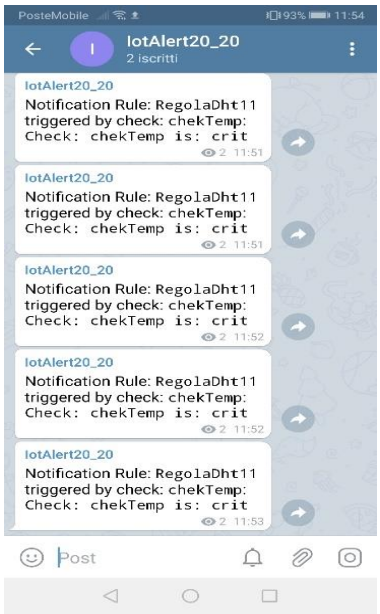
Risultati

Per testare il corretto funzionamento del tool per le fasi di visualizzazione, salvataggio e trasformazione si è provveduto a emulare tramite script sensori che producono un dato casuale. Il tool è stato anche testato tramite dati reali prodotti della board esp32 collegata a un sensore dht11 di temperatura e umidità. Lo sketch all'interno della board pubblica su un broker mqtt periodicamente i dati prodotti dal sensore.



Per testare ulteriormente il tool si sono visualizzati i dati attraverso la dashboard messa a disposizione da influx e attraverso il tool Grafana come visualizzato dalle immagini sottostanti.

Per quanto riguarda la parte degli alert tramite appositi script è possibile scegliere su quale canale avvertire il superamento di determinate soglie. Si sottolinea il fatto che per scegliere l'endpoint telegram tramite il tool Influx 2.0 bisogna richiamare apposite api tramite la libreria (3) in quanto l'interfaccia grafica non permette la possibilità di scegliere telegram come endpoint. Per testare il corretto funzionamento degli alert si è impostata la soglia di 19 gradi, se la temperatura scende sotto i 19 gradi un alert su un canale telegram viene inviato. Alternativamente è possibile eseguire una richiesta post che si occupa dell'invio su telegram della notifica.



Riferimenti

1. **Baroni, Andrea.** Codice sorgente Iot Bridge Platform. [Online] https://github.com/AndreaBaroni92/IoT-Bridge-Platform.

2. **Get started with Flux. [Online] https://docs.influxdata.com/influxdb/v2.0/query-data/get-starte.**

3. API Reference Documentation of InfluxDB v2 JavaScript Client . [Online] https://influxdata.github.io/influxdb-client-js/.