Applicazione IoT

Abbiamo visto come i dispositivi IoT abbiano risorse limitate, e di conseguenza gli sviluppatori hanno dovuto creare dei protocolli appositi per permettere la comunicazione tra cliet, servers e dispositivi IoT.

Come comunicano i dispositivi IoT?

Protocollo CoAP

Il protocollo CoAP è molto simile al protocollo HTTP, ma si basa su UDP. Di conseguenza l'affidabilità deve essere implementata a livello applicativo, invece che a livello di trasporto.

E' un tipo di comunicazione molto efficiente, anche grazie all'utilizzo della tecnica del **piggybacking**.

Protocollo MQTT

Il protocollo MQTT è completamente diverso sia da HTTP che da Coap. L'architettura di questa protocollo è composta da **publisher e subscriber**. Questo protocollo utilizza l'architettura TCP ed implementa l'affidabilotà sia a livello applicazione che a livello di trasporto. Abbiamo diversi livelli di affidabilità: QoS0, QoS1, QoS2.

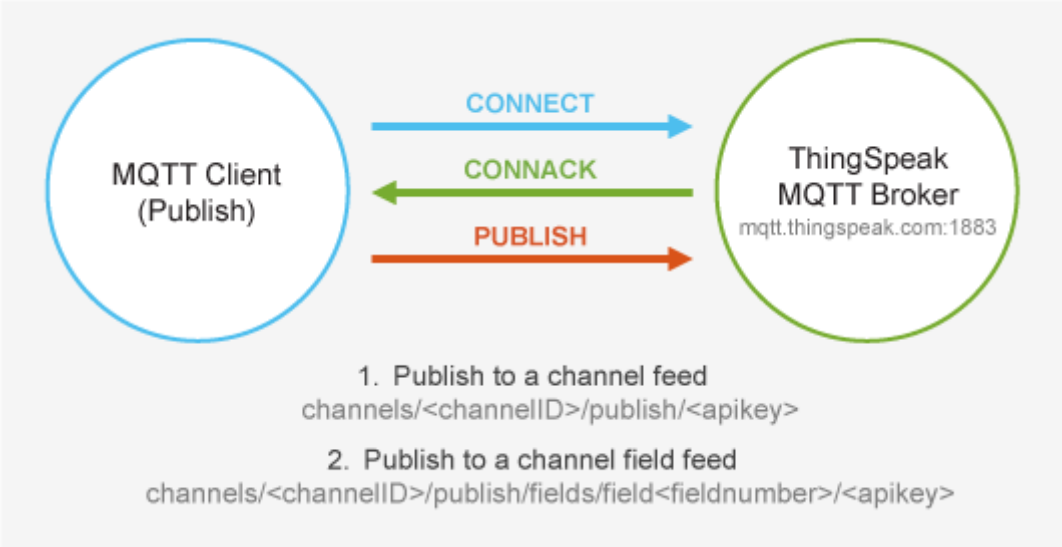
E' proprio questo protocollo che utilizza la nostra applicazione IoT.

ThingSpeak

ThinkSpeak è un software open source che permette agli users di comunicare con dispositivi IoT. Permette di facilitare l'accesso, raccoglimento e logging dei dati offrendo una **API** sia ai devices che ad applicazioni.

ThingSpeak ha da poco introdotto un **MQTT Broker** in modo che i devices possano inviare **messaggi** a ThingSpeak. Un messaggio potrebbe contenere la **temperatura corrente** in un officio controllato da un sensore.

ThingSpeak prende il messaggio e **salva il suo contenuto** all'interno di un **canale ThingSpeak**. Quando i dati sono presenti nel canale, si possono visualizzare in modo semplice o analizzare attraverso del codice MATLAB.



Il codice

Nel codice sono presenti diverse funzioni che ci permettono di leggere i sensori, ad esempio questa è una funzione che ci permette di leggere la temperatura da un sensore DHT.

```
void recTemp(){
  Serial.println("Reading temp...");
  temp = dht.readTemperature();   //reading temperature
  Serial.print("temp: ");
  Serial.println(temp);
}
```

Dopo aver letto i sensori, dobbiamo connetterci ad internet in modo da poter inviare dati al **broker** di ThingSpeak:

```
void connectToWiFi(){
    WiFi.begin(ssid, pass);                                    //Connect
to WiFi network
    Serial.print("Starting");
    while (WiFi.status() != WL_CONNECTED)
    {
      delay(500);
      Serial.print(".");
    }

    Serial.println();

    Serial.print("Connected, IP address: ");
    Serial.println(WiFi.localIP());
}
```

Una volta connessi al wifi, dobbiamo poter aggiornare i dati presenti sul server, usiamo quindi la funzione `updateThingSpeak()`:

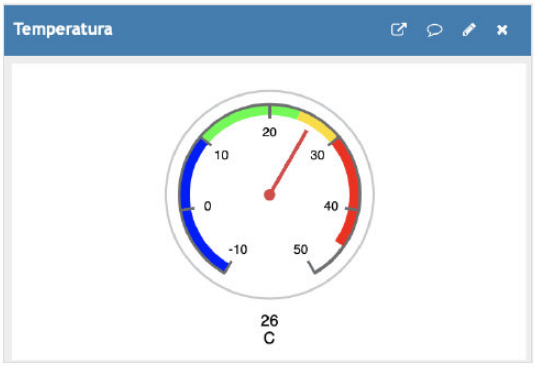
```
void updateThingSpeak ()                                       //Function
to post data to Thingspeak
{
  ThingSpeak.setField(1, temp);
  ThingSpeak.setField(2, hum);
  ThingSpeak.setField(3, percTemp);
  ThingSpeak.setField(4, windSpeed);

  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200)
  {
    Serial.println("Channel update successful.");
  }
  else
  {
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }
}
```

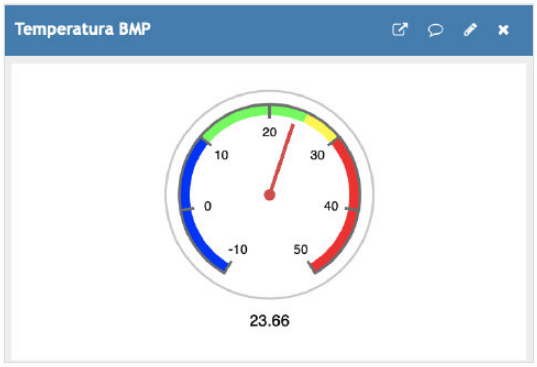
Sul canale ThingSpeak abbiamo diversi **fields**, ed ognuno fa riferimento ad un diverso sensore. Possiamo scrivere sul server grazie ad una **API key**.

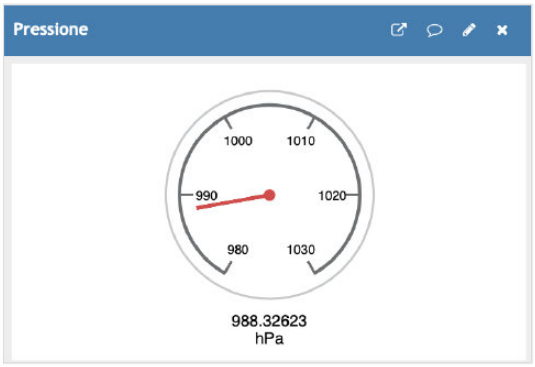
Infine, abbiamo un loop che ci permette di ciclare all'infinito e di leggere i sensori **ogni 10 secondi**

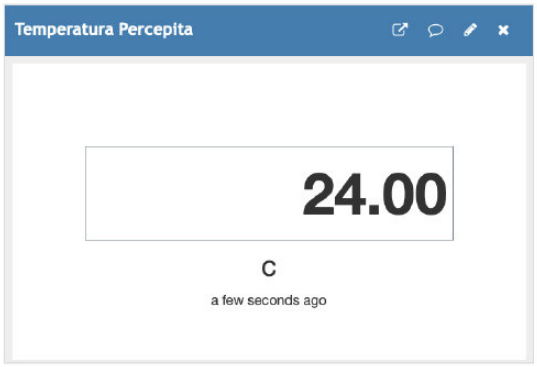
```
void loop() {
  if ((millis() - lastTime) > timerDelay) {
    // ############ WIFI OPS ########
    if(WiFi.status() != WL_CONNECTED){
      connectToWiFi();
    }

    Serial.println("Connecting to database");

    // ############ THINGSPEAK OPS ########
    ThingSpeak.begin(client);
 //Initialise ThingSpeak
    delay(10000);   //wait 10 secs

    Serial.println("Connected.");
    Serial.println("");
    Serial.println("Reading sensors...");


    recWindSpeed();

    recTemp();
    recHum();
    recPercTemp();

    //check if sensors are ok
    if(checkDHT11() && temp > -10 && temp < 50 && hum > -1 && hum < 100){
      Serial.println("Sending data...");
      updateThingSpeak();

      Serial.println("Sended");
    }else{
      Serial.println("Failed reading sensors.\nWebsite data not updated.");
    }

    lastTime = millis();
  }
}
```

L'interfaccia









Sul [Canale ThingSpeak] Possiamo visualizzare tutte le informazioni che abbiamo inviato al broker. ThingSpeak si occupa di salvare i dati all'interno dei propri server, in modo tale da permettere ai client di accedervi in ogni momento.

Possiamo anche importare/esportare i dati raccolti dai sensori: