

IoT con MQTT su ESP8266

Lo scopo di questa è utilizzare il protocollo MQTT per lo scambio di messaggi fra dispositivi IoT su ESP8266, programmabile dall'IDE di Arduino e con funzionalità per la connessione WiFi.

ESP8266 può quindi essere connesso direttamente ai sensori e agli attuatori e svolgere funzionalità di client MQTT.

1. Installazione delle librerie nel IDE di Arduino

a. Aggiungere in "File/Impostazioni/URL aggiuntive per il gestore di schede" il seguente URL:

https://arduino.esp8266.com/stable/package_esp8266com_index.json

- b. Aggiungere in "Strumenti/Scheda/Gestore schede" la scheda ESP8266; questo installerà, oltre al compilatore adatto, anche la libreria ESP8266WiFi
- c. Aggiungere in "Strumenti/Gestione librerie" la libreria PubSubClient
- d. Scegliere come scheda "Generic ESP8266 module"

2. Programma di esempio

Adattare e provare il seguente esempio:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// WiFi
const char *WIFI_SSID = "x";           // inserire i dati della rete WiFi
const char *WIFI_PASSWORD = "x";

// MQTT Broker
const char *MQTT_BROKER = " x.x.x.x"; // indirizzo IP del broker
const char *MQTT_USERNAME = "";       // se necessario
const char *MQTT_PASSWORD = "";
const int MQTT_PORT = 1883;

// topic
const char *INVIA = "miotopic/invia";
const char *RICEVI = "miotopic/ricevi";

// oggetti per wifi e mqtt
WiFiClient espClient;
PubSubClient client(espClient);

// setup
void setup() {

  Serial.begin(9600);
  Serial.println("Inizio");

  Serial.print("Connessione al WiFi..");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
```

```

        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("Connesso");

    // assegna un nome questo client
    String client_id = "esp8266-client-" + String(WiFi.macAddress());
    Serial.printf("%s in connessione al MQTT .." , client_id.c_str());
    client.setServer(MQTT_BROKER, MQTT_PORT);
    client.setCallback(callback);
    while (!client.connected()) {
        if (client.connect(client_id.c_str(), MQTT_USERNAME, MQTT_PASSWORD)) {
            Serial.println("Connesso al broker mqtt");
        } else {
            Serial.print("Fallito con codice di errore: ");
            Serial.println(client.state());
            Serial.println("Ritento");
            delay(2000);
        }
    }
    // pubblicazione di un primo messaggio
    client.publish(INVIA, "Attivo");

    // sottoscrizione
    client.subscribe(RICEVI);
}

// ciclo di invio messaggi
int n = 0;
void loop() {
    char s[20];
    n = n + 1;
    sprintf(s, "%d", n);
    client.publish(INVIA, s);
    delay(10000);
    client.loop();
}

// funzione di callback
void callback(char *topic, byte *payload, unsigned int length) {
    Serial.print("Arrivato un messaggio nel topic: ");
    Serial.println(topic);
    Serial.print("Messaggio:");
    for (int i = 0; i < length; i++) {
        Serial.print((char) payload[i]);
    }
    Serial.println();
    Serial.println("-----");
}

```

D1 Mini Pinout Diagram

