

ThingSpeak MQTT

Wie kann man die in der Cloud gespeicherten Daten sinnvoll nutzen?

Stand 06.09.2021

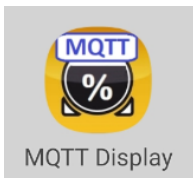
Mithilfe des WLAN Moduls und der bereits vorprogrammierten Software V1.xx.xx oder V2.xx.xx sendet der [Füllstandsmesser mit Drucksonde](#) seine Daten über das Internet an Ihren ThingSpeak Cloud Account. Bei einem Füllstandsmesser reicht es völlig aus, wenn Sie alle 15 Minuten aktuelle Daten senden lassen. Der Sendeintervall kann selbst angepasst werden.

Was macht man nun mit diesen Cloud Daten?

- >>> Diagramme der letzten Tage, Monate oder Jahre erstellen = geht natürlich
- >>> Die Daten der letzten Jahre miteinander vergleichen = geht auch
- >>> Die Daten in Ihr Excel exportieren = geht auch
- >>> Die Jahresregenmenge berechnen lassen = sollte mit MATLAB auch möglich sein

Was geht noch

- Ich habe eine kleine APP für Android Handys programmiert, mit der Sie den Füllstand Ihrer Zisterne an jedem Ort der Welt aufrufen können. Voraussetzung ist, Sie haben Handynet oder ein anderes nutzbares WLAN zur Verfügung.



Die passende APP bekommen Sie kostenfrei von mir.

Details / Ansichten in der APP (Hauptseite, Setup Channel, Setup Color)



b) ThingSpeak-Clouddaten über einen Raspberry Pi mittels Python abrufen und dann in eigenen Anwendungen beliebig weiter verarbeiten.

```
#!/usr/bin/env python
import urllib2, json, time
READ_API_KEY='FZXXXXXXXXXXXXLCLC'
CHANNEL_ID=1XXXXXX
def mqtt_read():
    conn = urllib2.urlopen("https://api.thingspeak.com/channels/%s/feeds/last.json?api_key=%s" % (CHANNEL_ID,READ_API_KEY))
    response = conn.read()
    data=json.loads(response)
    a = int(data['field1'])
    conn.close()
    return a

while (True) :
    wert = mqtt_read()
    print (wert)
    time.sleep(30)
```

Wenn man seine eigenen Daten (CHANNEL_ID und READ_API_Key) eingibt und diese Datei als „datenlesen.py“ speichert kann man sich über „python datenlesen.py“ den Füllstand sich alle 30 Sekunden (Zeile 16) ausgeben lassen.

c) ThingSpeak-Clouddaten über einen ESP8266 / ESP32 mittels freier Arduino-Software weiterverarbeiten.

```
/*
  24.02.2021 by Jens Dietrich
  - ThingSpeak mqtt read example
  - for private and public Channel
  - read only field1
  - Bord = ESP 8266 WEMOS D1
  */

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include "ThingSpeak.h" // load library for thingspeak
WiFiClient mqttclient;

unsigned long myChannelNumber = 1XXXXXX; // your channel
const char * myCounterReadAPIKey = "FZXXXXXXXXXXXXLCLC"; // your read API Key
const int FieldNumber = 1; // your field for read

char ssid_sta[30] = "xxxxxxxxxx"; // your SSID
char pass_sta[30] = "xxxxxxxxxxxxxxxx"; // your password

#define LED 2 // Port Wemos D4 = GPIO 2

void setup() {
  Serial.begin(115200);
  ThingSpeak.begin(mqttclient);
  WiFi.mode(WIFI_STA); // Wifi close
  WiFi.mode(WIFI_OFF); // Wifi off
  WiFi.begin(ssid_sta, pass_sta);
  pinMode(LED, OUTPUT); // LED pin is output
  digitalWrite(LED, LOW); // LED on, while wait for wifi
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
  }
  Serial.println();
  Serial.print("Connected, IP address: ");
  Serial.println(WiFi.localIP());
  digitalWrite(LED, HIGH); // LED off
}

void loop() {
  int wert = ThingSpeak.readIntField(myChannelNumber, FieldNumber, myCounterReadAPIKey);
  Serial.println(wert);

  int x = ThingSpeak.getLastReadStatus();
  if (x == 200) {
    Serial.println("Read successfull");
  }
  else {
    Serial.println("Problem reading channel. HTTP error code " + String(x));
  }
  delay(20000); // wait 20 seconds
}
```

Mit diesem Arduino Code-Schnipsel werden Ihre Clouddaten alle 20 Sekunden abgerufen und passend angezeigt.

Vorher ist die Bibliothek „ThingSpeak“ über Arduino, Werkzeuge und Bibliotheksverwalter einzubinden.

Der Weg erst einmal die Daten in der Cloud zu speichern und dann diese Daten wieder selbst abzurufen ist meiner Meinung nach sehr viel leichter, sicherer und besser, als die die Daten direkt aus dem eigenen Netzwerk abzuholen.

BITTE LESEN !!!

Die gezeigten Codezeilen sind nur als Anregungen zu verstehen und können von Ihnen beliebig genutzt und verändert werden. Ich werde keine „w...“ Fragen beantworten und schreibe nicht für Ihre Ideen die Software. Das können Sie mit etwas Interesse und Geduld selbst viel besser als ich.

Jens Dietrich
Bautzener Str. 9a
D-02977 Hoyerswerda
Tel.: +49 (0) 3571 6027653
Fax.: +49 (0) 3571 6027654

jd@icplan.de
<https://www.icplan.de>