**MQTT mit ESP8266 und ESP32**

Youtube Video: <https://www.youtube.com/watch?v=deIRY5NxXo4&t=28s>

**Arduino IDE:**

**Für ESP32 Datei – Voreinstellungen – Bordverwalter:**

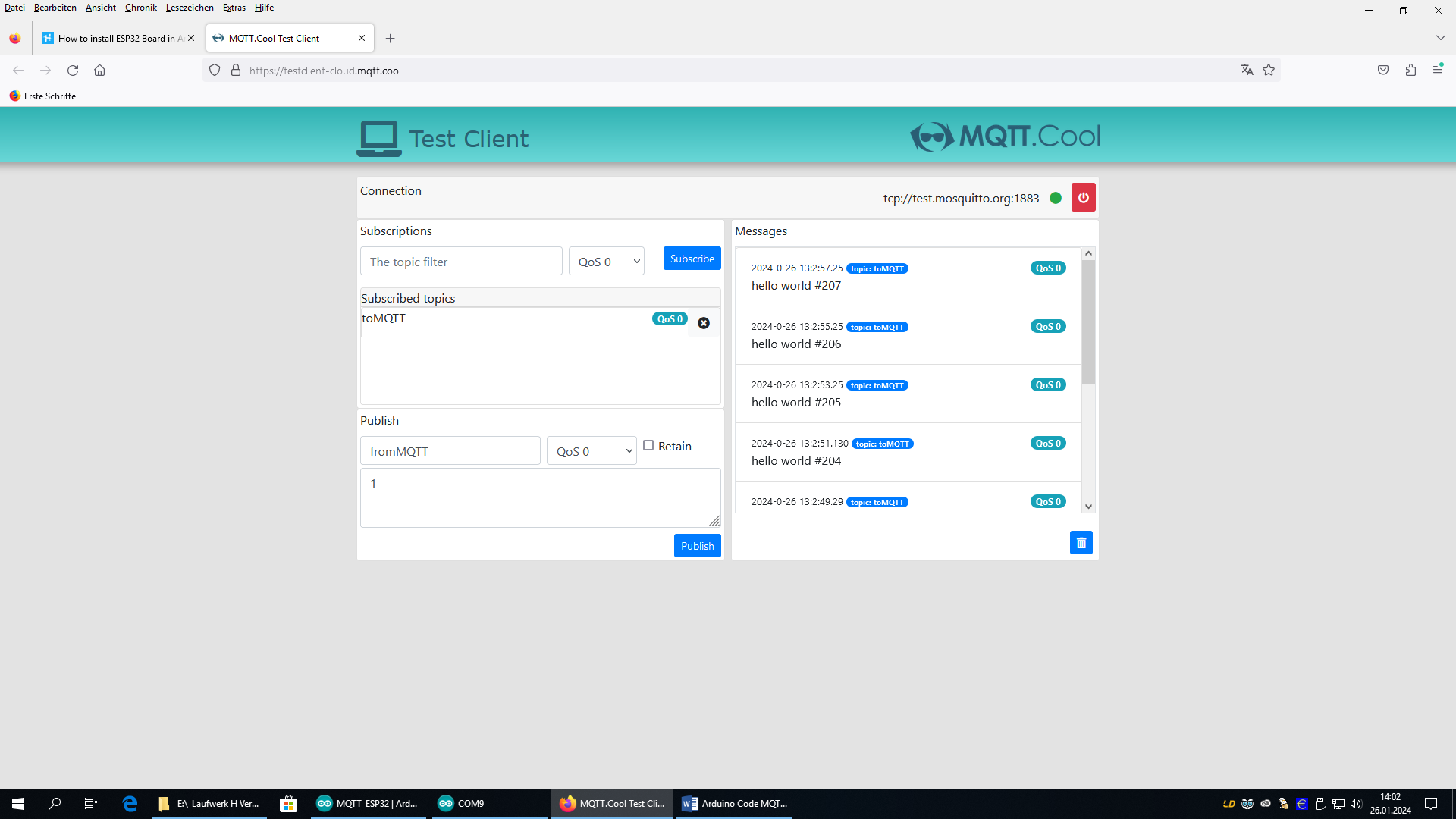
[**https://dl.espressif.com/dl/package\_esp32\_index.json**](https://dl.espressif.com/dl/package_esp32_index.json)

**Library einbinden:**

PubSubClient von Nick O‘ Leary

**Boardwahl: ESP32 Dev Module**

Mosquitto-Testseite: <https://testclient-cloud.mqtt.cool/>



**Mit ESP32 Mikrocontroller:**

**LED an Pin14**

/\*

 Basic ESP32 MQTT example

 This sketch demonstrates the capabilities of the pubsub library in combination

 with the ESP32 board/library.

 It connects to an MQTT server then:

  - publishes "hello world" to the topic "toMQTT" every two seconds

  - subscribes to the topic "fromMQTT", printing out any messages

    it receives. NB - it assumes the received payloads are strings not binary

  - If the first character of the topic "inTopic" is an 1, switch ON the LED,

    else switch it off

 It will reconnect to the server if the connection is lost using a blocking

 reconnect function. See the 'mqtt\_reconnect\_nonblocking' example for how to

 achieve the same result without blocking the main loop.

\*/

#include<WiFi.h>  //Für ESP32 notwendig

#include <PubSubClient.h>

#define LED 14

// Update these with values suitable for your network.

const char\* ssid = "HGS-WLAN-IoT";

const char\* password = "IoTHGSTG133";

const char\* mqtt\_server = "test.mosquitto.org";

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

int value = 0;

//----------------------------------------

void setup\_wifi()

{

  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED)

{

    delay(500);

    Serial.print(".");

  }

  randomSeed(micros());

  Serial.println("");

  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());

}

//--------------------------------------------------------

void callback(char\* topic, byte\* payload, unsigned int length)

{

  Serial.print("Message arrived [");

  Serial.print(topic);

  Serial.print("] ");

  for (int i = 0; i < length; i++)

  {

    Serial.print((char)payload[i]);

  }

  Serial.println();

  // Switch on the LED if an 1 was received as first character

  if ((char)payload[0] == '1')

  {

    digitalWrite(LED, HIGH);   // Turn the LED on

  }

  else

  {

    digitalWrite(LED, LOW);  // Turn the LED off

  }

}

//-------------------------------------------------------

void reconnect()

{

  // Loop until we're reconnected

  while (!client.connected())

{

    Serial.print("Attempting MQTT connection...");

    // Create a random client ID

    String clientId = "ESP8266Client-";

    clientId += String(random(0xffff), HEX);

    // Attempt to connect

    if (client.connect(clientId.c\_str()))

{

      Serial.println("connected");

      // Once connected, publish an announcement...

      client.publish("toMQTT", "hello world");

      // ... and resubscribe

      client.subscribe("fromMQTT");

    }

else

{

      Serial.print("failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 5 seconds");

      // Wait 5 seconds before retrying

      delay(5000);

    }

  }

}

//----------------------------------------------------

void setup()

{

  pinMode(LED,OUTPUT);

  Serial.begin(115200);

  setup\_wifi();

  client.setServer(mqtt\_server, 1883);

  client.setCallback(callback);

}

//-----------------------------------------------

void loop()

{

  if (!client.connected())

{

    reconnect();

  }

  client.loop();

  unsigned long now = millis();

  if (now - lastMsg > 2000)

{

    lastMsg = now;

    ++value;

    snprintf (msg, MSG\_BUFFER\_SIZE, "hello world #%ld", value);

    Serial.print("Publish message: ");

    Serial.println(msg);

    client.publish("toMQTT", msg);

  }

}

=====================================================================

**Mit ESP8266**

ESP8266 gewählt: NodeMCU 1.0 (ESP-12E Module)

Bei ESP8266 den Treiber CH340 zuvor Installieren!!!

<https://sparks.gogo.co.nz/ch340.html>

**Code mit ESP8266**

/\*

 Basic ESP8266 MQTT example

 This sketch demonstrates the capabilities of the pubsub library in combination

 with the ESP8266 board/library.

 It connects to an MQTT server then:

  - publishes "hello world" to the topic "outTopic" every two seconds

  - subscribes to the topic "inTopic", printing out any messages

    it receives. NB - it assumes the received payloads are strings not binary

  - If the first character of the topic "inTopic" is an 1, switch ON the ESP Led,

    else switch it off

 It will reconnect to the server if the connection is lost using a blocking

 reconnect function. See the 'mqtt\_reconnect\_nonblocking' example for how to

 achieve the same result without blocking the main loop.

 To install the ESP8266 board, (using Arduino 1.6.4+):

  - Add the following 3rd party board manager under "File -> Preferences -> Additional Boards Manager URLs":

       http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

  - Open the "Tools -> Board -> Board Manager" and click install for the ESP8266"

  - Select your ESP8266 in "Tools -> Board"

\*/

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// Update these with values suitable for your network.

const char\* ssid = "HGS-WLAN-IoT";

const char\* password = "IoTHGSTG133";

const char\* mqtt\_server = "test.mosquitto.org";

WiFiClient espClient;

PubSubClient client(espClient);

unsigned long lastMsg = 0;

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

int value = 0;

void setup\_wifi() {

  delay(10);

  // We start by connecting to a WiFi network

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

  randomSeed(micros());

  Serial.println("");

  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());

}

void callback(char\* topic, byte\* payload, unsigned int length) {

  Serial.print("Message arrived [");

  Serial.print(topic);

  Serial.print("] ");

  for (int i = 0; i < length; i++) {

    Serial.print((char)payload[i]);

  }

  Serial.println();

  // Switch on the LED if an 1 was received as first character

  if ((char)payload[0] == '1') {

    digitalWrite(BUILTIN\_LED, LOW);   // Turn the LED on (Note that LOW is the voltage level

    // but actually the LED is on; this is because

    // it is active low on the ESP-01)

  } else {

    digitalWrite(BUILTIN\_LED, HIGH);  // Turn the LED off by making the voltage HIGH

  }

}

void reconnect() {

  // Loop until we're reconnected

  while (!client.connected()) {

    Serial.print("Attempting MQTT connection...");

    // Create a random client ID

    String clientId = "ESP8266Client-";

    clientId += String(random(0xffff), HEX);

    // Attempt to connect

    if (client.connect(clientId.c\_str())) {

      Serial.println("connected");

      // Once connected, publish an announcement...

      client.publish("toMQTT", "hello world");

      // ... and resubscribe

      client.subscribe("fromMQTT");

    } else {

      Serial.print("failed, rc=");

      Serial.print(client.state());

      Serial.println(" try again in 5 seconds");

      // Wait 5 seconds before retrying

      delay(5000);

    }

  }

}

void setup() {

  pinMode(BUILTIN\_LED,OUTPUT);     // Initialize the BUILTIN\_LED pin as an output

  Serial.begin(115200);

  setup\_wifi();

  client.setServer(mqtt\_server, 1883);

  client.setCallback(callback);

}

void loop() {

  if (!client.connected()) {

    reconnect();

  }

  client.loop();

  unsigned long now = millis();

  if (now - lastMsg > 2000) {

    lastMsg = now;

    ++value;

    snprintf (msg, MSG\_BUFFER\_SIZE, "hello world #%ld", value);

    Serial.print("Publish message: ");

    Serial.println(msg);

    client.publish("toMQTT", msg);

  }

}