

```

DHT11.ino

#include "secrets.h"
#include <WiFiClientSecure.h>
#include <MQTTClient.h>
#include <ArduinoJson.h>
#include "WiFi.h"
#include <DHT.h>

#define DHTPIN 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// The MQTT topics that this device should publish/subscribe
#define AWS_IOT_PUBLISH_TOPIC "esp32/pub"
#define AWS_IOT_SUBSCRIBE_TOPIC "esp32/sub"

WiFiClientSecure net = WiFiClientSecure();
MQTTClient client = MQTTClient(256);

void connectAWS()
{
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    Serial.println("Connecting to Wi-Fi");

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    // Configure WiFiClientSecure to use the AWS IoT device credentials
    net.setCACert(AWS_CERT_CA);
    net.setCertificate(AWS_CERT_CRT);
    net.setPrivateKey(AWS_CERT_PRIVATE);

    // Connect to the MQTT broker on the AWS endpoint we defined earlier
    client.begin(AWS_IOT_ENDPOINT, 8883, net);

    // Create a message handler

```

```

client.onMessage(messageHandler);

Serial.print("Connecting to AWS IOT");

while (!client.connect(THINGNAME)) {
    Serial.print(".");
    delay(100);
}

if(!client.connected()){
    Serial.println("AWS IoT Timeout!");
    return;
}

// Subscribe to a topic
client.subscribe(AWS_IOT_SUBSCRIBE_TOPIC);

Serial.println("AWS IoT Connected!");
}

void publishMessage()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    StaticJsonDocument<200> doc;
    doc["time"] = millis();
    doc["Temp"] = t;
    doc["Hum"] = h;

    char jsonBuffer[512];
    serializeJson(doc, jsonBuffer); // print to client
    Serial.println(jsonBuffer);
    client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
}

void messageHandler(String &topic, String &payload) {
    Serial.println("incoming: " + topic + " - " + payload);

    // StaticJsonDocument<200> doc;
    // deserializeJson(doc, payload);

```

```
// const char* message = doc["message"];
}
```

```
void setup() {
    Serial.begin(115200);
    connectAWS();
    dht.begin();
}
```

```
void loop() {
    publishMessage();
    client.loop();
    delay(10000);
}
```

Secrets.h

```
#include <pgmspace.h>
```

```
#define SECRET
```

```
#define THINGNAME "DHT11_sensor"
```

```
const char WIFI_SSID[] = "iPhone (2)";
```

```
const char WIFI_PASSWORD[] = "987654321";
```

```
const char AWS_IOT_ENDPOINT[] =
```

```
"a26skg16wtvgxz-ats.iot.eu-central-1.amazonaws.com";//new code
```

```
//const char AWS_IOT_ENDPOINT[] =
```

```
"a26skg16wtvgxz-ats.iot.eu-north-1.amazonaws.com";
```

```
// Amazon Root CA 1
```

```
static const char AWS_CERT_CA[] PROGMEM = R"EOF(
```

```
-----BEGIN CERTIFICATE-----
```

```
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF
```

```
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBbWF6
```

```
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFoOTEL
```

```
MAkGA1UEBhMCVVMxZDZANBgNVBAoTBkFtYXpvcjEZMBcGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jl091ghYPl0hAEvraIthtOgQ3pOsqTQNroBvo3bSMgHFzZM
906II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrIqg0CAwEAAaNCMEAwdWYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFiQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIB3DQEBCwUA
A4IBAQCYY8jdaQZChGsV2USggNiMoruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjtt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
o/ufQJvtMVT8QtPHR8jrdkPSHca2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----
) EOF";
```

```
// Device Certificate
static const char AWS_CERT_CRT[] PROGMEM = R"KEY(
-----BEGIN CERTIFICATE-----
MIIDWTCCAkGgAwIBAgIUaQHZD9V6zvnJKiQLQ0XYIvBFwfkWdQYJKoZIhvcNAQEL
BQAwTTFLEkGA1UECwxCQW1hem9uIFdldiBTZXJ2aWNlcyBPPUftYXpvcjE5bjB2Og
SW5jLiBMPVNLyXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTEzMtIwMjA4MDU0
MFoXDTE5MTIzMtIzNTk1OVowHjEcmBoGA1UEAwTQVdTIElVVCBDZXJ0aWZpY2F0
ZTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBEBAJSdNKynYXTz8wc/0mcG
Fow3K+u+ycg/nR25KpRnBPWk2EcRLjA1334cbkXdeQBH+WY5zSXt38SxlkzAtwJj
i2RC9EtjQXpf0vm8hno+FQ60pUgXxiLjVlYb+o79Zg5h6mG4KZtf5oF35bevNq1N
d+/7MPpSkd2YkeEO5seL0aT5ZUKLK3gCsBJ0jQ1ckjofqluMDl9Dqx77XGg6ypWmb
Bfeh4iLP3nTnpJO/iKP2o1CVcwyn4mX8016rxg2aJhwMzY/GRAROvZzt0thN/XE
NrWKHWxyYZrar2wKI41W++DPj0RODw367GTHVuCR7MPCzZnS8HuapLzQQRGoFE3d
ip8CAwEAAaNgMF4wHwYDVR0jBBgwFoAU9SoGJvIMKSt8G3CZvGarRsglwf0wHQYD
VR0OBBYEFi2Cd7zNWq6daoFMJ1AXvn71tK+nMAwGA1UdEwEB/wQCMAAwDgYDVR0P
AQH/BAQDAgeAMA0GCSqGSIB3DQEBCwUAA4IBAQCcdhhgb8dKqnta7C7AMoKkoJvb
VoHsddjhHmZ4zjohoT5Fr4qzSDjLTminIz4XEST36NNqGaPcD2fk+Ie00R3xLq1t
9Ftw2rzujy8BCZyYf+qCgp50aFY7zVuVqHJYKRGHcLthsWHuaiWnDEyLnqwmI4Hr
m0rNSTBA4tZzPNAX7lhlHhv8fy3grDdRqU1TfOET6yL3ze2jf6VM7+kpbCMPvbca
sYQvP0NEIz1Gk4M8i7dEV5b7G4HlAlvd0A3jIbP0r5hG9I1Wjd+cc8KDNXFqwcw7
X7JpN+HLKe42KnIbKcRjIdEKpfNHHLvSFUNys+OIkk574ywbNyqdG9SKAE/U
-----END CERTIFICATE-----
```

```
)KEY";

// Device Private Key
static const char AWS_CERT_PRIVATE[] PROGMEM = R"KEY(
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAlJ00rKdhdPPzBz/SZwYWjDcr677JyD+dHbkqlGcE9aTYRxEu
MDXffhxuRd15AEf5bLnNJe3fxLGWTMC3AmOLZEL0S2NBel/S+byGej4VDrSlSBfG
IuNXVhv6jv1mDmHqYbgpm1/mgXflt682rU137/sw+lKR3ZiQQ7mx4vRpP1lQosre
AKXuPSNDVySOh+qW4wOX0OrHvtcaDrKlaZsF96HiIs/edOekk7+Io/ajUJVzDKfi
ZfzTXqvGDZomHAzNj8ZECug5Vn03S2E39cQ2tYodbHJhmtqvbAoJJVb74M+PRE4P
DfrsZMdW4JHsw8LNmdLwe5qkvNBBEagUTd2KnwIDAQABAoIBAEDBLw578E8RbLef
64HQeUqnT4/tFecXMNKeuBrpuJOAPGza45FGKBe/M51lny+sLNZfLzBzsaj26FKR
NizoSvho5CF1Ika8MiV3K/Ff07UQqWP8V+0jug5fKpHbf/iMWaAyqo50/u8y2v1q
cxyYQMZkYTbxe3Ve4J1h8Ey/yovLnQfH/V+XfUqPOq+qNXu6jeohuh3Y3d74Pqem
3+G9hBALUn2fw/zh1h8RMA239NAiRkj4FffZcKKsnq7fqXUcc9nrH0kBUZSzPrxk
u+4jwf3zIHt4bcdcd4z4c+JxQ2DVAshpQ51cVV1Ha+DfLZLL0/WG1c0fim/d6K40
TgAeCWEcGyEAw/ESvI8+Jh3ADvDukOomlc+/eWlabWYbNhP3YtQHU99CIh6537sI
8usEQR3G3DBT/G/ahWZTSUaPAk0T6zmZUG4xDrKSzJym7NhOr48yIjLrvmaYC3fG
Y0UsCirkY0q8K0IReXZBTY7eh9R0TLNj8DmcTs59RsQAQiuJNyIvNRMCGyEAwip8
BR4xhy5M45UJA0PB3/GrSK/CBUon9MM1HUj4A5btXy/r9BHslJrcMQzS6k2gQPlD
IoxqYwGnNJsJSoMn2k70QsFcQCucNNj0HwUkwAWG01z7o3zRm9uidRY8d8q/6dvn
Dz1ugbQNH7pvwdNlp0B9Gem9qZGbK4HG+IcM4cUCgYB8GZ+oVYEjWwxhajNvDR5z
pRS8rQFc8KKjaZnxuiC5+1lEivlmvms4j6kuJNHie6l15NIeU2TEsC+7YV4QSxbU
/jWL/YYEmeMPdyC/CI7TGrq86PW7raylCwaNmgt07kyniTzQFRnfKXG/zcurGKNf
bt2U/CtUOqd/QVuy06/aiQKBgQDBYilR6lrAcwOmuV7vdTcRTmEIAPOC2AqdDh/7
BJMU8hQkRu4XAbTAdPeDtSrlU2i4sI/C8aZU/pdlt9k3S0QBbX/5CIgMO/qB/G9a
boXJaTqxxkIZYEDMcB/rALCFV0TIIgSxfzzGE3Bt1GQqbdUtYv35WQNFNOBMQhDW
h+PhiQKBgGTZ4EH2khHH0EVjGaN5tAHGKQeRyZ9rBqySA0+GymEVk+5M0vY3e/Zu
scr3Z/MNiQDJ/F4US2r1uf5qKDU9Fw8nFhR/x+etP5W3Ek9Edu/smc7ale3iNl3W
ah2D/xaPM+LtmaAgfv77yZVLa8Ftqth5yYF54eYiQtU1BYwGlV0
-----END RSA PRIVATE KEY-----
)KEY";
```