// Example of the different modes of the X.509 validation options

// in the WiFiClientBearSSL object

//

// Jul 2019 by Taejun Kim at www.kist.ac.kr

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <time.h>

#include <ArduinoJson.h>

#include <Ticker.h>

Ticker ticker;

const char\* ssid = "Campus7\_Room4\_2.4GHz"; //와이파이 AP, 또는 스마트폰의 핫스판 이름

const char\* pass = "\*\*\*\*\*\*\*"; // 와이파이 비밀번호와이파이 AP, 또는 스마트폰의 핫스판 비번

const char \*thingId = "pzoneiot"; // 사물 이름 (thing ID)

const char \*host = "a21xxb3zewrt7o-ats.iot.ap-northeast-1.amazonaws.com"; // AWS IoT Core 주소

const char\* clientName = ""; // setup 함수에서 자동생성

String sChipID;

char cChipID[20];

char msg[100];

char msg1[100];

int count=0;

int tilt = 0;

// 사물 인증서 (파일 이름: xxxxxxxxxx-certificate.pem.crt)

const char cert\_str[] PROGMEM = R"EOF(

-----BEGIN CERTIFICATE-----

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

888888BQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi5jb20g

SW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIzMDIwNzA4MjU0

NFoXDTQ5MTIzMTIzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0

ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJeEHaLqe//7M44N/c1v

HK525MXOtqZSPU++2MQL5Yjs3ysWHXjMUnhgTm1q/ee3s+8JnRjiNsYg/ukZszku

zef1KPBTdFXsChtMlWJYT8U60iNmpsk/3WYkHDb2+6R+Uck/cIKf2mcuuepijjeu

6y3UnnU/jSh2s8EbxWzZ7aADFeLljnADe1HOcSEjUF5R+duVnuAX7aT2Iw6qvzPJ

3c2xdStwXQs20lsS7qlNUoh7r1gVw8GhnyTLvhhbrJb2kAcVzOs9hGW/eLBhwRVM

0G926i0QfQo+X0tKjb4XQlKTki4FSc45nQFRVw0i4zPdxrsNPaTb21+u87unmbEK

VdMCAwEAAaNgMF4wHwYDVR0jBBgwFoAUIYdlct2QmIvK2d/5D15IsbUB2yswHQYD

VR0OBBYEFHle6kanZAgvP7vAc3I2E7VB0h7LMAwGA1UdEwEB/wQCMAAwDgYDVR0P

AQH/BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IBAQAAaMdYStebFGvxYo6Ktjggw9qq

xWrnReWulo8LnFg7d/2Vfkgsioj0eOY1y3gSyWwsR75D2VEFfUqUCHHW/JJgzLVe

1/1t9hqjuQIR7fwaEe/ATjG2ZurduxQhPSqnbtp7bNjSeaSFua+hyzJMfbMqlWeU

HCRTyyVy5Y0cBQDcmtK/ke8CzO+DPs8CEUeo38wk7QsKNmlI+hEmrPEAzGi0GzB+

RARPGJWtO5SAzUASOlj809E/gTn3aOcCnyBWKpzDxFDeAt7k1XMEcoI50bCAkbMe

qxTJ4jXnsD/kFrKSi+DzFjI/pwtUDN4IjJ5V0ndakuCbN/GA8etMbVQku/8U

-----END CERTIFICATE-----

)EOF";

// 사물 인증서 프라이빗 키 (파일 이름: xxxxxxxxxx-private.pem.key)

const char key\_str[] PROGMEM = R"EOF(

-----BEGIN RSA PRIVATE KEY-----

MIIEpAIBAAKCAQEAl4Qdoup7//szjg39zW8crnbkxc62plI9T77YxAvliOzfKxYd

eMxSeGBObWr957ez7wmdGOI2xiD+6RmzOS7N5/Uo8FN0VewKG0yVYlhPxTrSI2am

yT/dZiQcNvb7pH5RyT9wgp/aZy656mKON67rLdSedT+NKHazwRvFbNntoAMV4uWO

cAN7Uc5xISNQXlH525We4BftpPYjDqq/M8ndzbF1K3BdCzbSWxLuqU1SiHuvWBXD

waGfJMu+GFuslvaQBxXM6z2EZb94sGHBFUzQb3bqLRB9Cj5fS0qNvhdCUpOSLgVJ

zjmdAVFXDSLjM93Guw09pNvbX67zu6eZsQpV0wIDAQABAoIBAEVGNnaiVwd2xt+Q

LhabUrXBzl9SX9gziazISTt5ZPzxsCd2+QqeAKbhAXqtjGFBcJc+AgG6i+iSdXoG

MhdW+vWu9YTL40jVntlUboiEcBRxDFr7OA6TkJp+sBfBXNScnLAyvTI7/BNpHb+4

R+XVp9lQeRKKd/OkMhN8qwtVqHubuWdCU7qDblYYoVEYv1f9qT6Cbm6alKCQLzb5

9RR6MwafYkbA5GdG97WmD8w/zjnxdl9gPwhj0YHM/VQDCOnanEug7WAh4F5dS75E

9Fl3m6f4vSg0mXnkq+Ux/6MJYmbHklN6idBnCO0qWTT+DAi+IoZmjcB8Ufdkudri

oQMvv2ECgYEAyTuJmBVy+MU1wKPEDo4LrBcUYBHqgD4ub6wy1PYLgMVLkZLzxFXB

R29mddh60uUdk6PEErUckvv+tHapTwWzYJd5/CHvkOcQenIn7JkRx//VGmoaRtkp

xq6lh2EYW53pkLyZhRbW/KTWhiILS5yF6Fu11OrSOt1PrNpfYA7KUr0CgYEAwMCv

SbtjOMD12ospMIPaKmVMRF3wtMPLs3XNE7VB55pNRp1YuH1Q6pp+mPz6mR4FoM5n

6yPNYN0eG0qMj46RFZ1+jcLrl9ZTl8rHbRZStMfmqckqnuovPKH+ava0A07pA4Bq

hvAs/toCIE+a8xYTzLMgsdV2PwvmoXrBRYLNm88CgYEAsdzj4v6iXmvA9QJK2iqG

+pZzte5f/PX5CU+EpmLLDAhWoNboixLZARDf2S18ckd8evB276oRFYGSQWPYhjYd

tDRu54H/Q25YTEH2M9OfuczQSMabbx+nIytet2gEpYXP2dQ5RT9IyonP60LQBQbV

/OiCsSzUoJ7YO4nQXxDzl2kCgYEArFOYY3ME4lihdSG1UYqQIhcueHEEIphqJafM

ETKcGVP6godqKobfMN+vD1gdM4eDwKDIh5EarN8Da6Lax/3N+GU2H8eEmdskITFJ

2l160+Vkc6dGU41LbgnmS/sENXD3IgpNMxbHI5IFFJ4XCf1S9k+vi04syXD6YKQ+

5sZqzEECgYBwtXobM9GK9QttVp4NVq8ewUwq5zYaIDpyiYer2sVPZaVnfbS4PjDf

gqs5bWdSXRboX/8IQVHIFcH86XHfJ3eHuizUTBYcJgbmGKngGiNrJ4y7TogtyM5Q

3WxqN85p4HGN8kUQauneO7kVSX//td3I4j13GXSqgM+YhgPVpXZl+w==

-----END RSA PRIVATE KEY-----

)EOF";

// Amazon Trust Services(ATS) 엔드포인트 CA 인증서 (서버인증 > "RSA 2048비트 키: Amazon Root CA 1" 다운로드)

const char ca\_str[] PROGMEM = R"EOF(

-----BEGIN CERTIFICATE-----

ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBB//

b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL

MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv

b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj

ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM

9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw

IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6

VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L

93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm

jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC

AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA

A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI

U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1Bjjt/msv0tadQ1wUs

N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv

o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU

5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy

rqXRfboQnoZsG4q5WTP468SQvvG5

-----END CERTIFICATE-----

)EOF";

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();

}

X509List ca(ca\_str);

X509List cert(cert\_str);

PrivateKey key(key\_str);

WiFiClientSecure wifiClient;

PubSubClient client(host, 8883, callback, wifiClient); //set MQTT port number to 8883 as per //standard

void reconnect() {

// Loop until we're reconnected

while (!client.connected()) {

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

// Attempt to connect

if (client.connect(thingId)) {

Serial.println("connected");

// Once connected, publish an announcement...

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

// ... and resubscribe

client.subscribe("inTopic");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

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

char buf[256];

wifiClient.getLastSSLError(buf,256);

Serial.print("WiFiClientSecure SSL error: ");

Serial.println(buf);

// Wait 5 seconds before retrying

delay(5000);

}

}

}

// Set time via NTP, as required for x.509 validation

void setClock() {

configTime(3 \* 3600, 0, "pool.ntp.org", "time.nist.gov");

Serial.print("Waiting for NTP time sync: ");

time\_t now = time(nullptr);

while (now < 8 \* 3600 \* 2) {

delay(500);

Serial.print(".");

now = time(nullptr);

}

Serial.println("");

struct tm timeinfo;

gmtime\_r(&now, &timeinfo);

Serial.print("Current time: ");

Serial.print(asctime(&timeinfo));

}

void setup() {

// ticker.attach(3, tick); //0.1 초도 가능

//ticker.detach();

//---------------------my chip edit

//이름 자동으로 생성

sChipID=String(ESP.getChipId(),HEX);

sChipID.toCharArray(cChipID,sChipID.length()+1);

clientName=&cChipID[0];

Serial.println(clientName);

// client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

//------------------my chip end

Serial.begin(115200);

Serial.setDebugOutput(true);

Serial.println();

Serial.println();

// We start by connecting to a WiFi network

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, pass);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

wifiClient.setTrustAnchors(&ca);

wifiClient.setClientRSACert(&cert, &key);

Serial.println("Certifications and key are set");

setClock();

//client.setServer(host, 8883);

client.setCallback(callback);

}

long lastMsg = 0;

//char msg[50];

int value = 0;

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

long now = millis();

if (now - lastMsg > 1000) {

lastMsg = now;

++value;

tilt = analogRead(A0);

Serial.println(tilt);

delay(100);

String smsg = "{ \"sen\_id\" : \"";

smsg += sChipID;

smsg += "\", \"inc\" : ";

smsg += tilt;

smsg += " }";

client.publish("outTopic", smsg.c\_str());

}

}