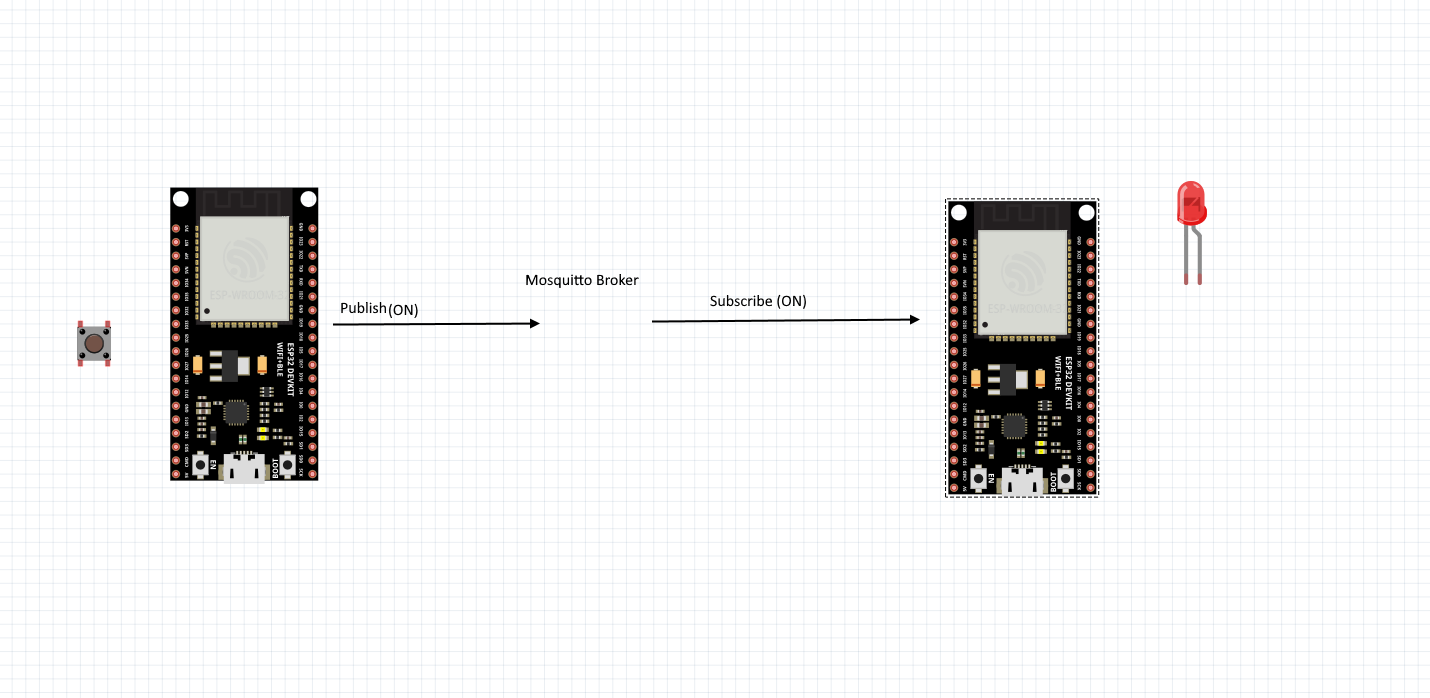
**ESP32 MQTT**



MQTT server :  
name : test.mosquitto.org

topic: Led\_Test

Publish:  
A circuit board with wires connected to it

Description automatically generated

#include <Arduino.h>

#include <PubSubClient.h>

#include <WiFi.h>

#define button\_pin 33

#define led\_pin 5

const char\* ssid = "seamk-titelab";

const char\* password = "Sula289tite";

WiFiClient wifiClient;

PubSubClient mqttClient(wifiClient);

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

void initWiFi() {

  WiFi.mode(WIFI\_STA);

  WiFi.begin(ssid, password);

  Serial.print("Connecting to WiFi ..");

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

    Serial.print('.');

    delay(1000);

  }

  Serial.println(WiFi.localIP());

}

void reconnect() {

  while (!mqttClient.connected()) {

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

    if (mqttClient.connect("arduinoClient")) {

      Serial.println("connected");

    } else {

      Serial.print("failed, rc=");

      Serial.print(mqttClient.state());

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

      delay(5000);

    }

  }

}

void setup()

{

  Serial.begin(9600);

  pinMode(button\_pin, INPUT\_PULLUP);

  initWiFi();

  mqttClient.setServer(mqtt\_server, 1883);

  delay(1500);

}

void loop()

{

  if (!mqttClient.connected()) {

    reconnect();

  }

  Serial.println(digitalRead(button\_pin));

  if (!digitalRead(button\_pin))

  {

    mqttClient.publish("Led\_Test", "On");

    Serial.println("On");

  }

  else{

    mqttClient.publish("Led\_Test", "Off");

    Serial.println("Off");

  }

  //mqttClient.publish("Led\_Test", "On");

  //Serial.println("julkaistu");

  delay(1000);

}

Subcribe:  
A circuit board with wires and wires

Description automatically generated

#include <Arduino.h>

#include <PubSubClient.h>

#include <WiFi.h>

// WiFi credentials and MQTT server details

const char\* ssid = "seamk-titelab";

const char\* password = "Sula289tite";

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

// GPIO pin where the LED is connected

#define led\_pin 5

WiFiClient wifiClient;

PubSubClient mqttClient(wifiClient);

void initWiFi() {

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

Serial.print("Connecting to WiFi ..");

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

  Serial.print('.');

  delay(1000);

}

Serial.println(WiFi.localIP());

}

void reconnect() {

while (!mqttClient.connected()) {

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

  // Attempt to connect

  if (mqttClient.connect("espClient")) {

    Serial.println("connected");

    // Subscribe to the topic

    mqttClient.subscribe("Led\_Test");

  } else {

    Serial.print("failed, rc=");

    Serial.print(mqttClient.state());

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

    delay(5000);

  }

}

}

// Callback function that will be executed when a message is received on the subscribed topic

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

Serial.print("Message arrived in topic: ");

Serial.println(topic);

Serial.print("Message:");

String message;

for (int i = 0; i < length; i++) {

  message += (char)payload[i];

}

Serial.println(message);

Serial.println("-----------------------");

// Turn the LED ON or OFF based on the message content

if (message == "On") {

  digitalWrite(led\_pin, HIGH); // Turn the LED on

} else if (message == "Off") {

  digitalWrite(led\_pin, LOW); // Turn the LED off

}

}

void setup() {

Serial.begin(9600);

pinMode(led\_pin, OUTPUT);

initWiFi();

mqttClient.setServer(mqtt\_server, 1883);

mqttClient.setCallback(callback);

}

void loop() {

if (!mqttClient.connected()) {

  reconnect();

}

mqttClient.loop();

}