Ex 6. ESP32 exchange data using MQTT

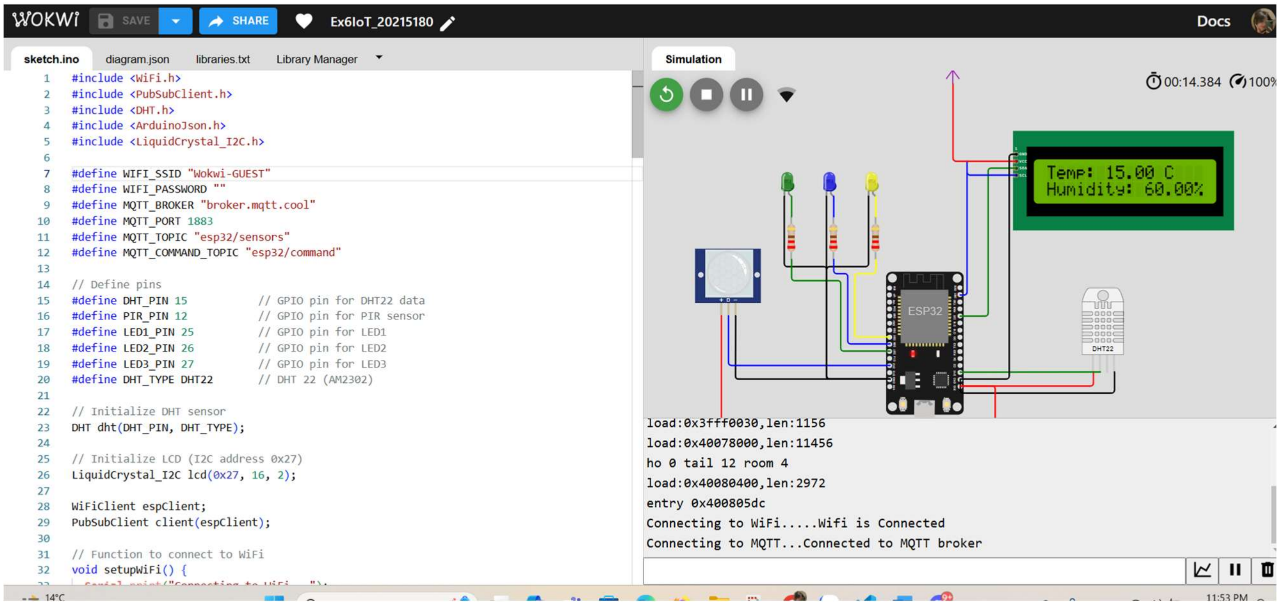
Tran Thi Lan Anh 20215180 – ICT02 K66

Write a program for ESP32 to send and receive data via MQTT

1. Source code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <LiquidCrystal_I2C.h>

#define WIFI_SSID "Wokwi-GUEST"
#define WIFI_PASSWORD ""
#define MQTT_BROKER "broker.mqtt.cool"
#define MQTT_PORT 1883
#define MQTT_TOPIC "esp32/sensors"
#define MQTT_COMMAND_TOPIC "esp32/command"

// Define pins
#define DHT_PIN 15           // GPIO pin for DHT22 data
#define PIR_PIN 12           // GPIO pin for PIR sensor
#define LED1_PIN 25          // GPIO pin for LED1
#define LED2_PIN 26          // GPIO pin for LED2
#define LED3_PIN 27          // GPIO pin for LED3
#define DHT_TYPE DHT22       // DHT 22 (AM2302)

// Initialize DHT sensor
DHT dht(DHT_PIN, DHT_TYPE);

// Initialize LCD (I2C address 0x27)
LiquidCrystal_I2C lcd(0x27, 16, 2);

WiFiClient espClient;
PubSubClient client(espClient);

// Function to connect to WiFi
void setupWiFi() {
  Serial.print("Connecting to WiFi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Wifi is Connected");
}

// Callback function for handling incoming MQTT messages
void callback(char* topic, byte* payload, unsigned int length) {
  String message = "";
  for (int i = 0; i < length; i++) {
    message += (char)payload[i];
  }

  if (String(topic) == MQTT_COMMAND_TOPIC) {
    DynamicJsonDocument doc(1024);
    DeserializationError error = deserializeJson(doc, message);
    if (error) {
      Serial.println("Failed to parse JSON");
      return;
    }

    // Control LEDs based on received JSON
    int led1 = doc["led1"];
    int led2 = doc["led2"];
    int led3 = doc["led3"];

    Serial.print("LED1: ");
    Serial.print(led1);
    Serial.print(" LED2: ");
    Serial.print(led2);
    Serial.print(" LED3: ");
    Serial.println(led3);

    digitalWrite(LED1_PIN, led1);
    digitalWrite(LED2_PIN, led2);
    digitalWrite(LED3_PIN, led3);
  }
}

// Function to send sensor data via MQTT
void sendSensorData() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  bool motionDetected = digitalRead(PIR_PIN);

  if (isnan(temperature) || isnan(humidity))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Display on LCD
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature);
  lcd.print(" C");
  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.print(humidity);
  lcd.print("%");

  DynamicJsonDocument doc(1024);
  doc["temperature"] = temperature;
  doc["humidity"] = humidity;
  doc["motion"] = motionDetected;

  char jsonBuffer[256];
  serializeJson(doc, jsonBuffer);

  client.publish(MQTT_TOPIC, jsonBuffer);
}

void setup() {
  Serial.begin(115200);

  pinMode(PIR_PIN, INPUT);
  pinMode(LED1_PIN, OUTPUT);
  pinMode(LED2_PIN, OUTPUT);
  pinMode(LED3_PIN, OUTPUT);

  // Start DHT sensor
  dht.begin();

  // Connect to WiFi
  setupWiFi();

  // Initialize LCD
  lcd.init();
  lcd.backlight();

  // Setup MQTT client
  client.setServer(MQTT_BROKER, MQTT_PORT);
  client.setCallback(callback);

  // Connect to MQTT broker
  while (!client.connected()) {
    Serial.print("Connecting to MQTT...");
    if (client.connect("ESP32_Client")) {
      Serial.println("Connected to MQTT broker");
      // Subscribe to command topic
      client.subscribe(MQTT_COMMAND_TOPIC);
    } else {
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      delay(2000);
    }
  }

  lcd.setCursor(0, 0);
  lcd.print("Initializing...");
  delay(2000);
  lcd.clear();
}

void loop() {
  // Maintain MQTT connection
  client.loop();

  // Send sensor data every 10 seconds
  sendSensorData();
  delay(10000); // Delay for 10 seconds before sending again
}
```

2. Link wokwi: https://wokwi.com/projects/417982406546533377
3. Demo:



```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <ArduinoJson.h>
#include <LiquidCrystal_I2C.h>

#define WIFI_SSID "Wokwi-GUEST"
#define WIFI_PASSWORD ""
#define MQTT_BROKER "broker.mqtt.cool"
#define MQTT_PORT 1883
#define MQTT_TOPIC "esp32/sensors"
#define MQTT_COMMAND_TOPIC "esp32/command"

// Define pins
#define DHT_PIN 15           // GPIO pin for DHT22 data
#define PIR_PIN 12           // GPIO pin for PIR sensor
#define LED1_PIN 25          // GPIO pin for LED1
#define LED2_PIN 26          // GPIO pin for LED2
#define LED3_PIN 27          // GPIO pin for LED3
#define DHT_TYPE DHT22       // DHT 22 (AM2302)

// Initialize DHT sensor
DHT dht(DHT_PIN, DHT_TYPE);

// Initialize LCD (I2C address 0x27)
LiquidCrystal_I2C lcd(0x27, 16, 2);

WiFiClient espClient;
PubSubClient client(espClient);

// Function to connect to WiFi
void setupWiFi() {
```

```
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
Connecting to WiFi.....Wifi is Connected
Connecting to MQTT...Connected to MQTT broker
```