

SloT →HW 3 →MQTT using ESP32

Parisa Toumari

99101857

1. Connecting ESP32 to the broker:

First, I opened a MQTT example of ESP-IDF extension in VSCode, so that all the needed configurations are set and I don't have to do it manually.

Then I replaced the example code with the actual code we need to use for the HW.

Summary of the code:

This code initializes Wi-Fi on an ESP32, connects to a specified Wi-Fi network, and then initializes and starts an MQTT client to publish a message to a broker. The Wi-Fi and MQTT events are handled by respective event handler functions. The MQTT broker used in this example is "mqtt.eclipseprojects.io", and the published message is "Hello! This is Parisa with 99101857, Enjoying HW3 of SloT course:))".

```
static void wifi_event_handler(void *event_handler_arg,
esp_event_base_t event_base, int32_t event_id, void *event_data)
```

This function handles Wi-Fi events such as connection, disconnection, and obtaining an IP address.

```
void wifi_connection()
```

This function initializes and configures the Wi-Fi connection, setting the SSID and password, and connects to the Wi-Fi network.

The laptop, and the ESP32 have to be in the same internet network; so we connect both of them to my mobile hotspot:

```
wifi_config_t wifi_configuration = {
    .sta = {
        .ssid = "Paris's iPhone",
        .password = "parispass"}};
```

```
static esp_err_t mqtt_event_handler_cb(esp_mqtt_event_handle_t
event)
```

This function handles MQTT events such as connection, disconnection, subscription, unsubscription, publication, received data, and errors.

```
static void mqtt_event_handler(void *handler_args, esp_event_base_t base, int32_t event_id, void *event_data)
```

This function is a wrapper around the MQTT event handler callback and dispatches events received from the event loop.

```
static void mqtt_app_start(void)
```

This function initializes the MQTT client with the broker's URI and registers the MQTT event handler.

```
void app_main(void)
```

This is the main function that initializes NVS flash, establishes a Wi-Fi connection, waits for a brief period, and then starts the MQTT application.

```
esp_mqtt_client_config_t mqtt_cfg = {  
    .uri = "mqtt://mqtt.eclipseprojects.io",  
};
```

This sets up the MQTT broker's URI for the MQTT client configuration.

```
esp_mqtt_client_handle_t client = esp_mqtt_client_init(&mqtt_cfg);  
esp_mqtt_client_register_event(client, ESP_EVENT_ANY_ID,  
mqtt_event_handler, client);  
esp_mqtt_client_start(client);
```

The MQTT client is initialized, the event handler is registered, and the client is started to establish a connection with the MQTT broker.

```
vTaskDelay(2000 / portTICK_PERIOD_MS);  
printf("WIFI was initiated .....\\n");
```

This introduces a delay after Wi-Fi initiation and prints a message indicating that the Wi-Fi was initiated.

```
ESP-IDF 4.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-94df9c5339738bd38b5e758c3b44ff47 - python.exe "C:\Espressif\frameworks\esp-idf...
I (698) wifi_init: WiFi IRAM OP enabled
I (698) wifi_init: WiFi RX IRAM OP enabled
I (708) phy_init: phy_version 4771,450c73b,Aug 16 2023,11:03:10
I (788) wifi:mode : sta (48:e7:29:96:bc:0c)
I (788) wifi:enable tsf
WiFi connecting ...
I (808) wifi:new:<11,0>, old:<1,0>, ap:<255,255>, sta:<11,0>, prof:1
I (808) wifi:state: init -> auth (b0)
I (848) wifi:state: auth -> assoc (0)
I (848) wifi:state: assoc -> run (10)
I (898) wifi:connected with Moh, aid = 3, channel 11, BW20, bssid = ea:a9:73:27:ab:26
I (898) wifi:security: WPA2-PSK, phy: bgn, rssi: -48
I (898) wifi:pm start, type: 1

WiFi connected ...
I (928) wifi:<ba:addr>idx:0 (ifx:0, ea:a9:73:27:ab:26), tid:0, ssn:0, winSize:64
I (968) wifi:AP's beacon interval = 102400 us, DTIM period = 2
I (1898) esp_netif_handlers: sta ip: 192.168.145.182, mask: 255.255.255.0, gw: 192.168.145.126
WiFi got IP ...

WiFi was initiated .....
I (2798) MQTT_TCP: Other event id:7
I (3618) MQTT_TCP: MQTT_EVENT_CONNECTED
I (4048) MQTT_TCP: MQTT_EVENT_SUBSCRIBED, msg_id=45470
I (4458) MQTT_TCP: MQTT_EVENT_DATA

TOPIC=my_topic
DATA=Hello! This is Parisa with 99101857, Enjoying HW3 of SIoT course:))
I (4868) MQTT_TCP: MQTT_EVENT_PUBLISHED, msg_id=14868
```

```
Administrator: Command Prompt - mosquitto_sub -t my_topic -h mqtt.eclipseprojects.io
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd ..

C:\Windows>cd ..

C:\>cd "Program Files"

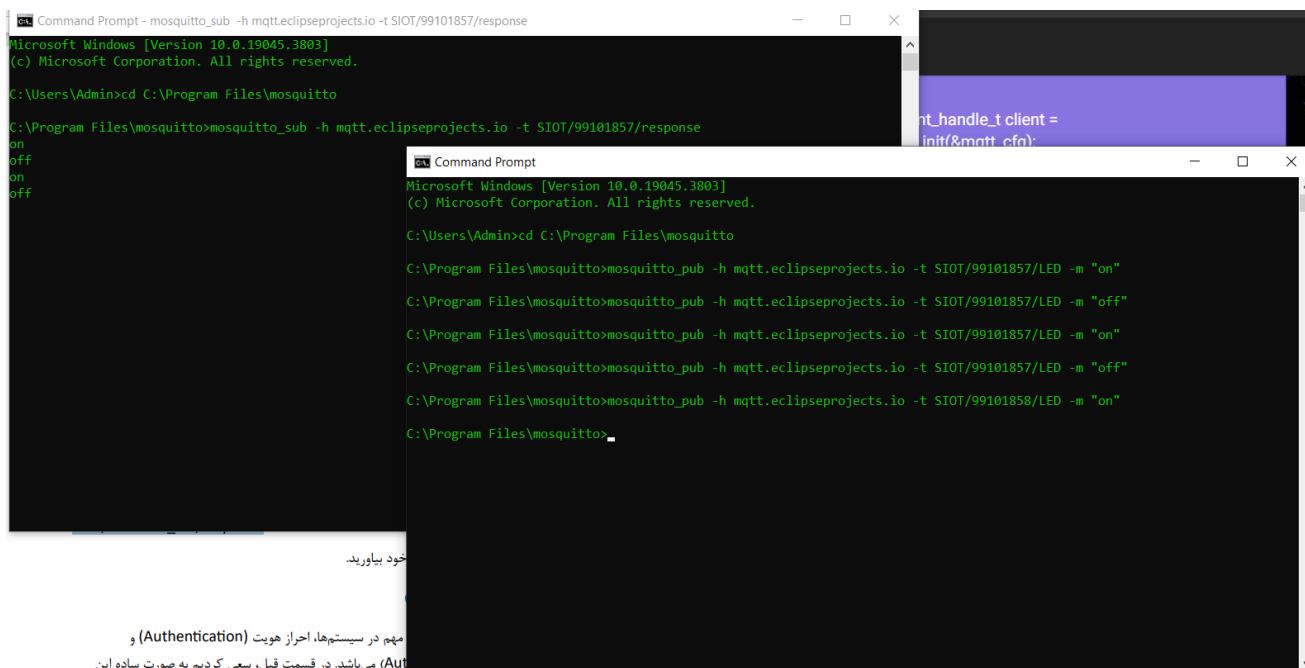
C:\Program Files>cd mosquitto

C:\Program Files\mosquitto>mosquitto_sub -t my_topic -h mqtt.eclipseprojects.io
Hello! This is Parisa with 99101857, Enjoying HW3 of SIoT course:))
Hello! This is Parisa with 99101857, Enjoying HW3 of SIoT course:))
```

2. Controlling LED:

This ESP32 application subscribes to the MQTT topic `SIOT/<student_id>/LED` and controls an LED based on the received messages ("on" or "off"). It also publishes a response message to the topic `SIOT/<student_id>/response`.

Also, I filmed the process and it's attached to my HW file.



3. Controlling LED using Authentication:

In this part, We use the same mosquitto configurations as we used in HW2 to authenticate users.

We need to add these 2 lines so that it would be possible for only a specific user to control the LED:

```
.username = "user1",  
.password = "1234",
```

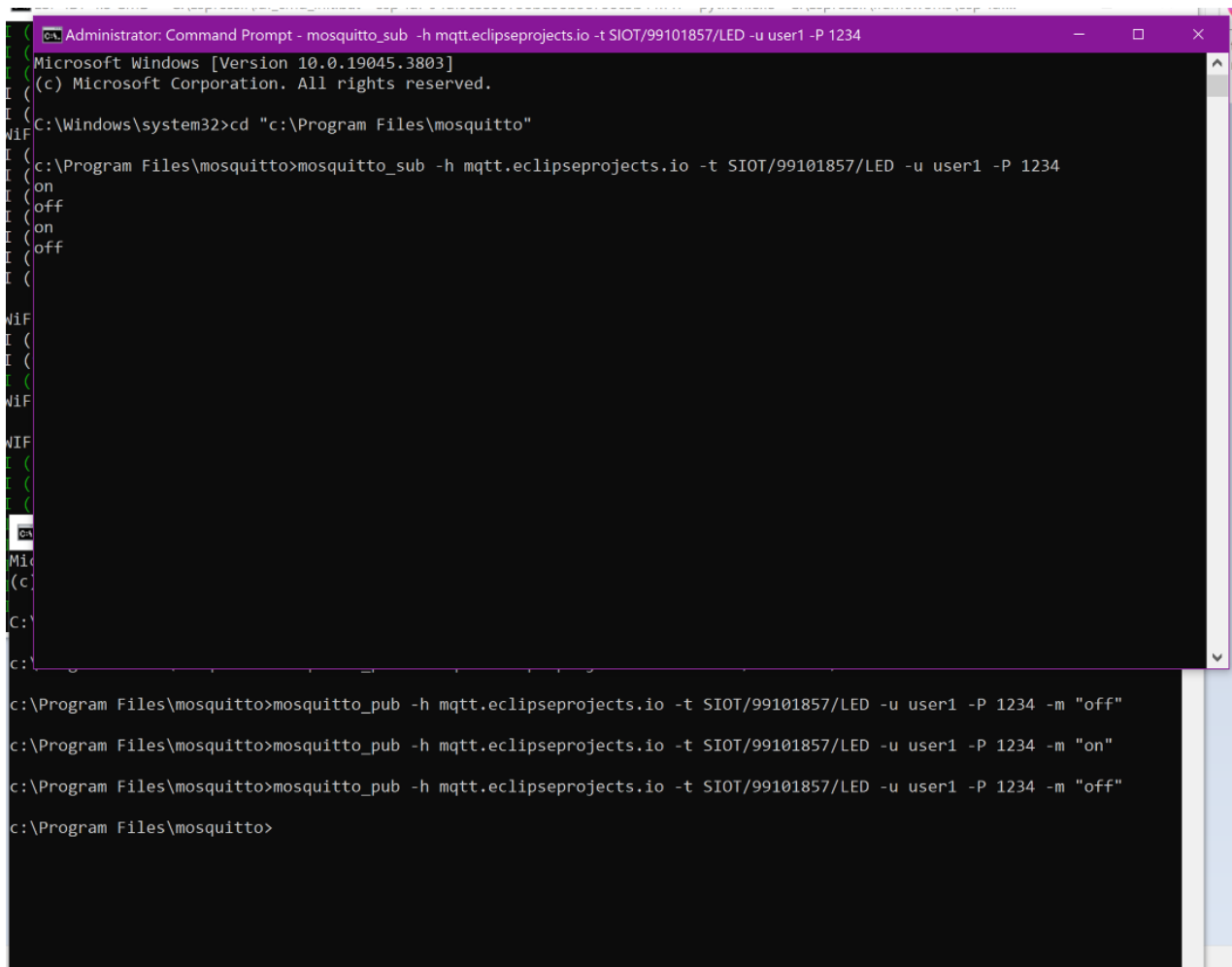
Also, I filmed the process and it's attached to my HW file.

Other configurations in the mosquitto.conf:

```
allow_anonymous false
password_file C:\Program Files\mosquitto\passwd
listener 1883
```

The “passwd” file:

```
user1:$7$101$abgRIRXWkc06Xg1f$4v4YWlaYCr7vii3hl0jhRKVsiO10YnPj1jF4s5PS5HDdiitOUwc8JER55jiC6RCzYunyE3rthEiMyFDNIO6plw
```



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - mosquitto_sub -h mqtt.eclipseprojects.io -t SIOT/99101857/LED -u user1 -P 1234". The window content is as follows:

```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd "c:\Program Files\mosquitto"

c:\Program Files\mosquitto>mosquitto_sub -h mqtt.eclipseprojects.io -t SIOT/99101857/LED -u user1 -P 1234
on
off
on
off

c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99101857/LED -u user1 -P 1234 -m "off"
c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99101857/LED -u user1 -P 1234 -m "on"
c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99101857/LED -u user1 -P 1234 -m "off"
c:\Program Files\mosquitto>
```