



HW3 SIOT

MOHAMMAD GHAFOURIANAHAHASSANPOUR

STUDENT NUMBER: 99106493

DR SIAVASH AHMADI

connecting to broker:

```
I (786) phy_init: phy_version 4771,450c730,Aug 16 2023,11:03:10
I (786) wifi:mode : sta (48:e7:29:96:bc:0c)
I (786) wifi:enable tsf
mmmaI (786) wifi:enable tsf
Wifi connecting ...
[VeI (806) wifi:new:<11,0>, old:<1,0>, ap:<255,255>, sta:<11,0>, prof:1
poraI (806) wifi:state: init -> auth (b0)
I (806) wifi:state: auth -> assoc (0)
32>cI (816) wifi:state: assoc -> run (10)
I (856) wifi:connected with Moh, aid = 3, channel 11, BW20, bssid = ea:a9:73:27:ab:26
mosqI (856) wifi:security: WPA2-PSK, phy: bgn, rssi: -31
I (866) wifi:pm start, type: 1
ker
Wifi connected ...
I (886) wifi:<ba-addr>idx:0 (ifx:0, ea:a9:73:27:ab:26), tid:0, ssn:0, winSize:64
mosqI (936) wifi:AP's beacon interval = 102400 us, DTIM period = 2
I (1866) 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 (2796) MQTT_TCP: Other event id:7
I (3606) MQTT_TCP: MQTT_EVENT_CONNECTED
I (4026) MQTT_TCP: MQTT_EVENT_SUBSCRIBED, msg_id=54365
I (4426) MQTT_TCP: MQTT_EVENT_DATA

TOPIC=my_topic
DATA=Hello! This is Mammad Ghafourian with 99106493, Enjoying HW3 of SIoT course:))
I (4836) MQTT_TCP: MQTT_EVENT_PUBLISHED, msg_id=15498
```

```
C:\WINDOWS\system32\cmd.exe - mosquitto_sub -h mqtt.eclipseprojects.io -t my_topic
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto_sub -h mqtt.eclipseprojects.io -t my_topic
Hello! This is Mammad Ghafourian with 99106493, Enjoying HW3 of SIoT course:))
Hello! This is Mammad Ghafourian with 99106493, Enjoying HW3 of SIoT course:))
```

First of all, in main function we should initial Wi-Fi and then and connect to a Wi-Fi and after we connected to Wi-Fi we have to run the MQTT protocol. First we call `nvs_flash_init` function to flash the esp then we call the `wi-fi_connection` and we stay for 2 seconds waiting to connect and be sure everything is fine, in the end the `mqtt_app_start` has been called to use MQTT protocol for subscribing and publishing the message.

```

}

void app_main(void)
{
    nvs_flash_init();
    wifi_connection();

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

    mqtt_app_start();
}

```

In `wifi_connection` function, the esp32 is going to connect to our Wi-Fi, the Wi-Fi name is "Moh" and its password is "uoar3287", and we log the Wi-Fi event with `wifi_event_handler` and we register the Wi-Fi and IP.

```

void wifi_connection()
{
    // 1 - Wi-Fi/LwIP Init Phase
    esp_netif_init(); // TCP/IP initiation s1.1
    esp_event_loop_create_default(); // event loop s1.2
    esp_netif_create_default_wifi_sta(); // WiFi station s1.3
    wifi_init_config_t wifi_initiation = WIFI_INIT_CONFIG_DEFAULT();
    esp_wifi_init(&wifi_initiation); // s1.4
    // 2 - Wi-Fi Configuration Phase
    esp_event_handler_register(WIFI_EVENT, ESP_EVENT_ANY_ID, wifi_event_handler, NULL);
    esp_event_handler_register(IP_EVENT, IP_EVENT_STA_GOT_IP, wifi_event_handler, NULL);
    wifi_config_t wifi_configuration = {
        .sta = {
            .ssid = "Moh",
            .password = "uoar3287"}};
    esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_configuration);
    // 3 - Wi-Fi Start Phase
    esp_wifi_start();
    // 4- Wi-Fi Connect Phase
    esp_wifi_connect();
}

```

```

static void wifi_event_handler(void *event_handler_arg, esp_event_base_t event_base, int32_t event_id, void *event_data)
{
    switch (event_id)
    {
        case WIFI_EVENT_STA_START:
            printf("Wifi connecting ... \\n");
            break;
        case WIFI_EVENT_STA_CONNECTED:
            printf("Wifi connected ... \\n");
            break;
        case WIFI_EVENT_STA_DISCONNECTED:
            printf("Wifi lost connection ... \\n");
            break;
        case IP_EVENT_STA_GOT_IP:
            printf("Wifi got IP ... \\n\\n");
            break;
        default:
            break;
    }
}

```

All we need in this code is to connect to a specific uri and after the connection the mqtt client protocol has been started. In `mqtt_event_handler_cb` we handle the events. This is where our subscription and publishing command is launched, in case of what event is, the algorithm is set up. In this assignment after the esp32 connected to mqtt protocol first we subscribe to the topic and then we published the message: "Hello! This is Mammad Ghafourian with 99106493, Enjoying HW3 of SIoT course:))". Yet other events are not important and we just report the state with LOG.

```
static void mqtt_app_start(void)
{
    esp_mqtt_client_config_t mqtt_cfg = {
        .uri = "mqtt://mqtt.eclipseprojects.io",
    };
    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);
}
```

```
static void mqtt_event_handler(void *handler_args, esp_event_base_t base, int32_t event_id, void *event_data)
{
    ESP_LOGD(TAG, "Event dispatched from event loop base=%s, event_id=%d", base, event_id);
    mqtt_event_handler_cb(event_data);
}
```

```
static esp_err_t mqtt_event_handler_cb(esp_mqtt_event_handle_t event)
{
    esp_mqtt_client_handle_t client = event->client;
    switch (event->event_id)
    {
        case MQTT_EVENT_CONNECTED:
            ESP_LOGI(TAG, "MQTT_EVENT_CONNECTED");
            esp_mqtt_client_subscribe(client, "my_topic", 0);
            esp_mqtt_client_publish(client, "my_topic", "Hello! This is Mammad Ghafourian with 99106493, Enjoying HW3 of SIoT course:)) ", 0, 1, 0);
            break;
        case MQTT_EVENT_DISCONNECTED:
            ESP_LOGI(TAG, "MQTT_EVENT_DISCONNECTED");
            break;
        case MQTT_EVENT_SUBSCRIBED:
            ESP_LOGI(TAG, "MQTT_EVENT_SUBSCRIBED, msg_id=%d", event->msg_id);
            break;
        case MQTT_EVENT_UNSUBSCRIBED:
            ESP_LOGI(TAG, "MQTT_EVENT_UNSUBSCRIBED, msg_id=%d", event->msg_id);
            break;
        case MQTT_EVENT_PUBLISHED:
            ESP_LOGI(TAG, "MQTT_EVENT_PUBLISHED, msg_id=%d", event->msg_id);
            break;
        case MQTT_EVENT_DATA:
            ESP_LOGI(TAG, "MQTT_EVENT_DATA");
            printf("TOPIC=%.*s\r\n", event->topic_len, event->topic);
            printf("DATA=%.*s\r\n", event->data_len, event->data);
            break;
        case MQTT_EVENT_ERROR:
            ESP_LOGI(TAG, "MQTT_EVENT_ERROR");
            break;
        default:
            ESP_LOGI(TAG, "Other event id:%d", event->event_id);
            break;
    }
    return ESP_OK;
}
```

Controlling LED:

In controlling LED first the addresses that we should respond and receive message has been defined:


```
static const char *TAG = "MQTT_TCP";
#define LED_PIN GPIO_NUM_2
#define STUDENT_ID "99106493"
#define MQTT_TOPIC_LED "SIOT/" STUDENT_ID "/LED"
#define MQTT_TOPIC_RESPONSE "SIOT/" STUDENT_ID "/response"
```

And the only part that it should have changed is in Wi-Fi connection handler when the event says Wi-Fi has been connected.

```
static esp_err_t mqtt_event_handler_cb(esp_mqtt_event_handle_t event)
{
    esp_mqtt_client_handle_t client = event->client;
    switch (event->event_id) {
        case MQTT_EVENT_CONNECTED:
            ESP_LOGI(TAG, "MQTT_EVENT_CONNECTED");
            esp_mqtt_client_subscribe(client, MQTT_TOPIC_LED, 0);
            break;
        case MQTT_EVENT_DATA:
            ESP_LOGI(TAG, "MQTT_EVENT_DATA");
            if (strcmp(event->topic, MQTT_TOPIC_LED, event->topic_len) == 0) {
                if (strcmp(event->data, "on", event->data_len) == 0) {
                    gpio_set_level(LED_PIN, 1); // Turn on LED
                    strcpy(message, "on");
                } else if (strcmp(event->data, "off", event->data_len) == 0) {
                    gpio_set_level(LED_PIN, 0); // Turn off LED
                    strcpy(message, "off");
                }
                esp_mqtt_client_publish(client, MQTT_TOPIC_RESPONSE, message, strlen(message), 0, 0);
            }
            break;
        case MQTT_EVENT_DISCONNECTED:
            // ...
    }
}
```

After esp connected to MQTT, it subscribes to led topic and then in event data we check the condition that the data which has been received equals to "on" or "off" and therefore turn the LED on or off and next publishes the message in response address.

```
I (789) WiFi:enable tcp
WiFi connecting ...
I (859) wifi:new:<1,0>, old:<1,0>, ap:<255,255>, sta:<1,0>, prof:1
I (869) wifi:state: init -> auth (b0)
I (869) wifi:state: auth -> assoc (0)
I (879) wifi:state: assoc -> run (10)
I (919) wifi:connected with Moh, aid = 2, channel 1, BW20, bssid = ea:a9:73:27:ab:26
I (919) wifi:security: WPA2-PSK, phy: bgn, rssi: -47
I (919) wifi:pm start, type: 1

WiFi connected ...
I (949) wifi:<ba-add>idx:0 (ifx:0, ea:a9:73:27:ab:26), tid:0, ssn:0, winSize:64
I (959) wifi:AP's beacon interval = 102400 us, DTIM period = 2
I (1929) 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 (2799) gpio: GPIO[2] InputEn: 0 OutputEn: 0 OpenDrain: 0 Pullup: 1 Pulldown: 0 Intr:0
I (2799) MQTT_TCP: Other event id:7
I (3539) MQTT_TCP: MQTT_EVENT_CONNECTED
I (3939) MQTT_TCP: MQTT_EVENT_SUBSCRIBED, msg_id=50857
I (207529) MQTT_TCP: MQTT_EVENT_DATA
I (257289) MQTT_TCP: MQTT_EVENT_DATA
I (285959) MQTT_TCP: MQTT_EVENT_DATA
I (300609) MQTT_TCP: MQTT_EVENT_DATA
```

```
Administrator: Command Prompt
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>net start mosquitto
The requested service has already been started.

For more help is available by typing NET HELPMSG 2182.

C:\Program Files\mosquitto>
```

```
Command Prompt - mosquitto_sub -h mqtt.eclipseprojects.io -t SIOT/99106493/response
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Program Files\mosquitto>
The syntax of the command is incorrect.

C:\Users\Admin>cd C:\Program Files\mosquitto

C:\Program Files\mosquitto>mosquitto_sub -h mqtt.eclipseprojects.io -t SIOT/99106493/response
on
off
```

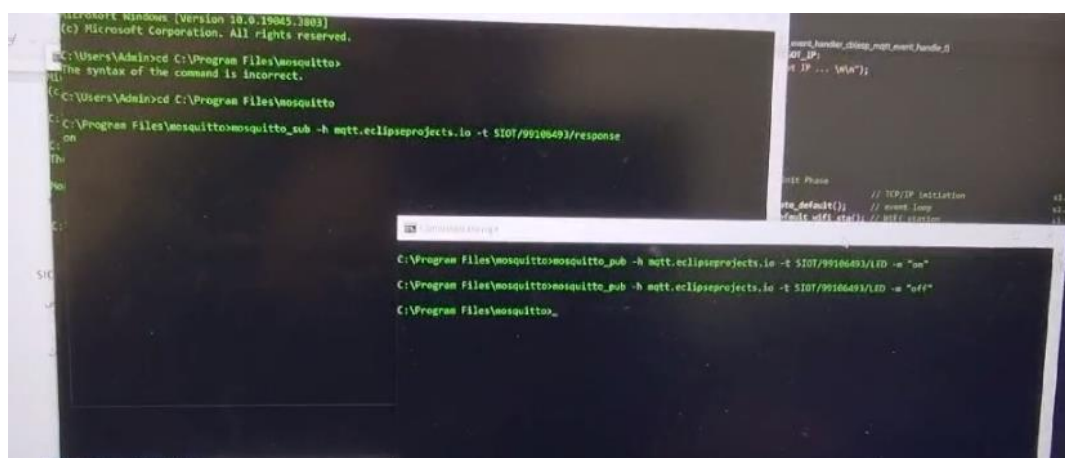
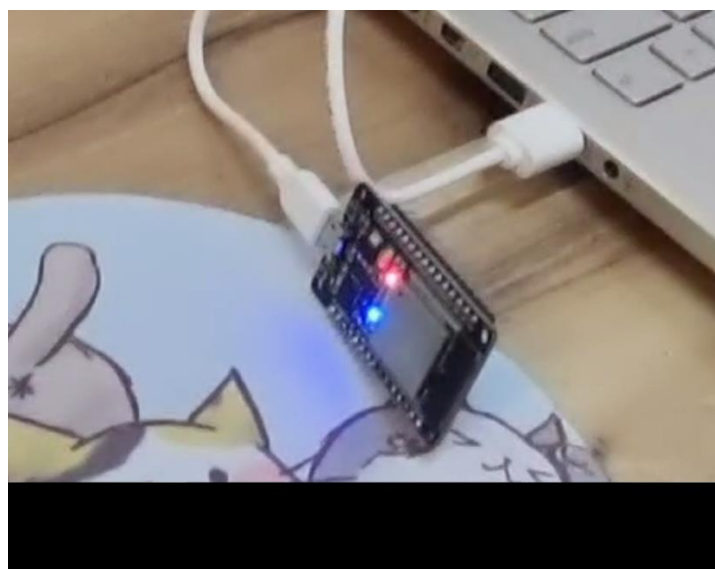
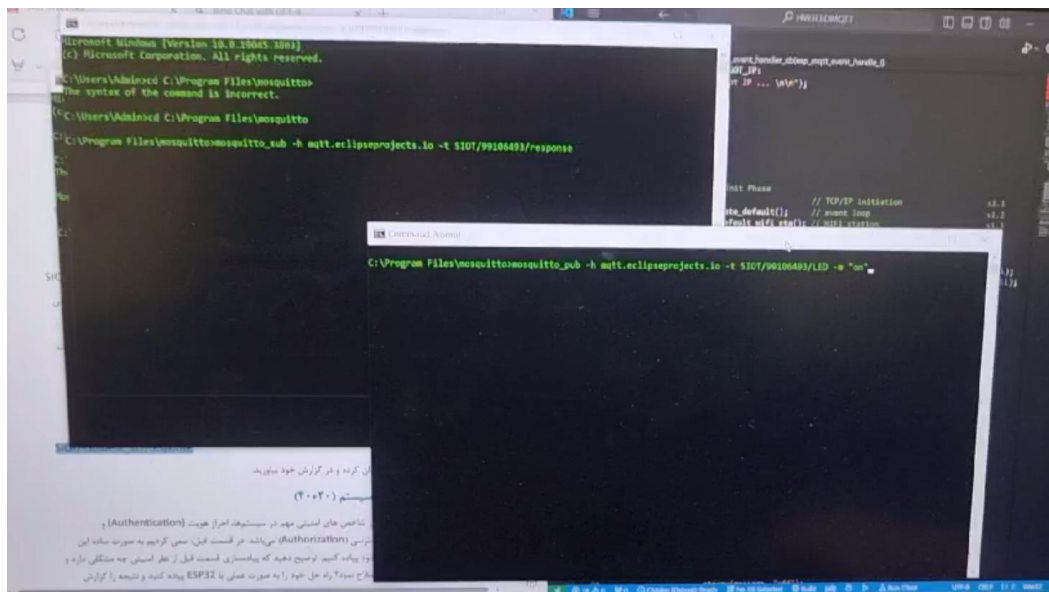
```
Command Prompt
C:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -m "on"
C:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -m "off"
C:\Program Files\mosquitto>
```

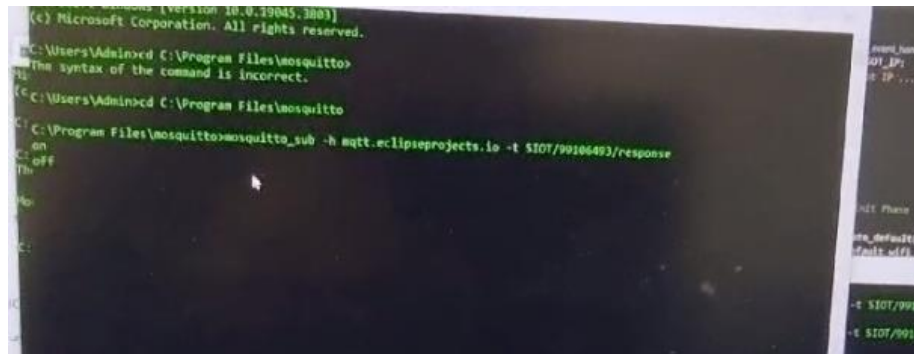
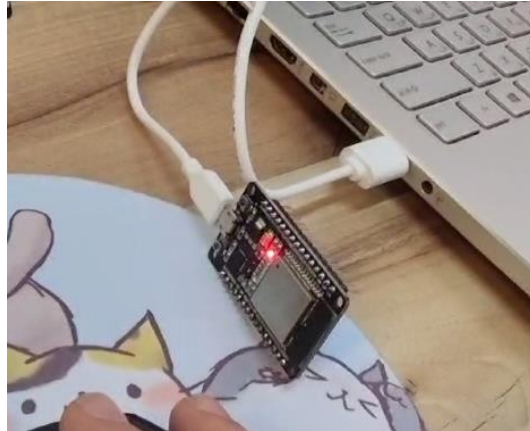
Student_id/response

ان کرده و در گزارش خود بیاورید.

سیستم (۴۰+۲۰)

از شاخص های امنیتی مهم در سیستمها، احراز هویت (Authentication) و
تربسی (Authorization) می باشد. در قسمت قبل، سعی کردیم به صورت ساده ا
تود پیاده کنیم. توضیح دهید که پیاده سازی قسمت قبل از نظر امنیتی چه مشکلی
سلاج نموده؟ راه حل خود را به صورت عملی با ESP32 ساده کنید و نتیجه را گزارش





Authentication:

In this section so as previous assignment we define password file with “mosquitto_passwd -c “passwordfile.pwd” mammad 1234” and then the esp32 should sign in to the topic with username password. The only change that we have to make in code is to make change in client configuration.

```
static void mqtt_app_start(void)
{
    esp_mqtt_client_config_t mqtt_cfg = {
        .uri = "mqtt://mqtt.eclipseprojects.io",
        .username = "mammad",
        .password = "1234",
    };
    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);
}
```



```
c:\Program Files\mosquitto>mosquitto -c mosquitto.conf -v
1704545915: mosquitto version 2.0.18 starting
1704545915: Config loaded from mosquitto.conf.
1704545915: Opening ipv6 listen socket on port 1883.
1704545915: Opening ipv4 listen socket on port 1883.
1704545915: mosquitto version 2.0.18 running
```

```
ESP-IDF 4.3 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-94df9c5339738bd38b5e758c3b44ff47 - python.exe "C:\Espressif\frameworks\esp-idf..."
Administrator: Command Prompt - mosquitto_sub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -u mammad -P 1234
c:\Program Files\mosquitto>mosquitto_sub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -u mammad -P 1234
I (on
I (off
WIF (on
I (off
I (
I (
I (
I (
I (
I (
WIF (
I (
I (
I (
WIF (
WIF (
I (
I (
Administrator: Command Prompt
c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -u mammad -P 1234 -m "on"
c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -u mammad -P 1234 -m "off"
c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -u mammad -P 1234 -m "on"
c:\Program Files\mosquitto>mosquitto_pub -h mqtt.eclipseprojects.io -t SIOT/99106493/LED -u mammad -P 1234 -m "off"
c:\Program Files\mosquitto>
```

