M480 WiFi MQTT 用戶端

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	本範例使用 M480 UART 驅動 ESP8266，實現 MQTT 用戶端功能
BSP 版本	M480 Series BSP CMSIS V3.04.000
開發平台	NuMaker-IoT-M487 v1.3

1 功能介紹

1.1 簡介

本範例程式碼使用M480系列晶片，通過UART1驅動ESP8266 WiFi模組，實現MQTT用戶端功能。當ESP8266模組連接上無線存取點後，開始建立與MQTT代理人的連線，並且每10秒發佈NuMaker-IoT-M487板子上SW2(GPIO F11)和SW3(GPIO G5)按鍵狀態的訊息。

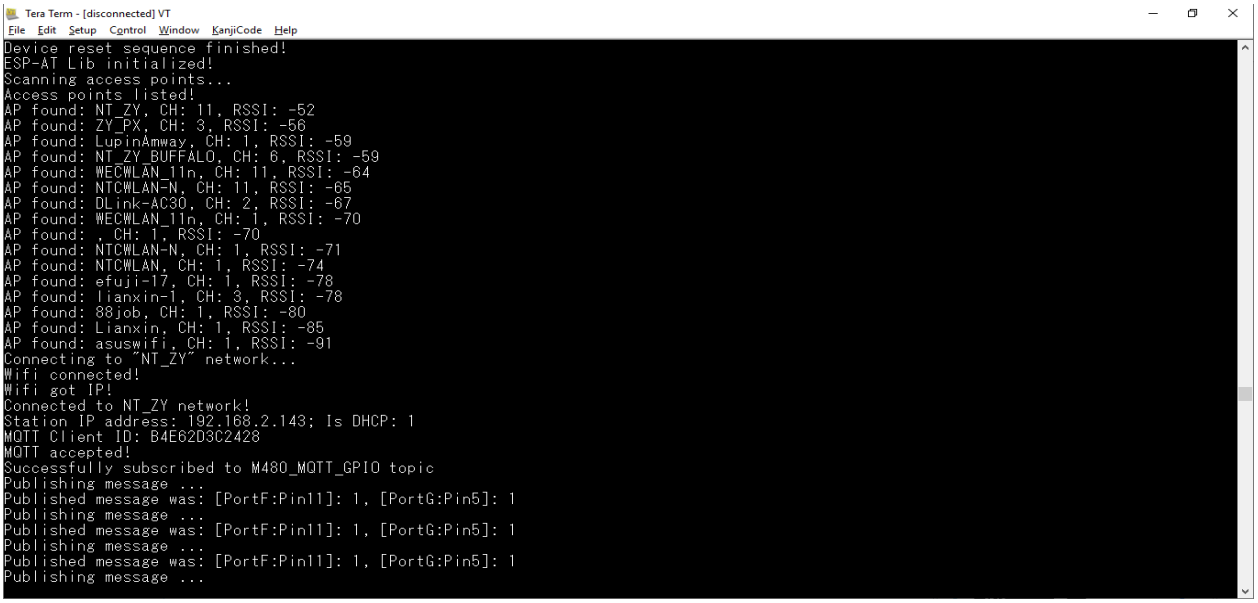
1.2 原理

範例程式核心使用了ESP-AT Lib函式庫。ESP-AT Lib是一個AT命令解析器，用於使用AT命令與ESP8266 WiFi模組通信。ESP-AT Lib提供了Station和Access point無線網路管理API，並提供了BSD socket相似的API，方便應用程式建立主從式架構的網路應用。

MQTT是一種基於"發佈(Publisher)/訂閱(Subscriber)"機制的訊息傳輸協定。一個MQTT系統由服務器和通信的用戶端所組成，而服務器通常稱為"代理人(Broker)"。用戶端可以是訊息的發佈者，也可以是訂閱者。

在範例程式裡，展示的是用戶端訊息的發佈者，固定會發佈主題為"M480_MQTT_GPIO"的訊息到代理人端。代理人端使用的是"test.mosquitto.org"服務器。最後在Android手機上使用MQTT Dashboard或者在iOS裝置上使用MQTTool在做為訂閱者，來接收主題為"M480_MQTT_GPIO"的訊息。

1.3 執行結果



2 代碼介紹

設置連接無線存取點的 SSID 和 Key：

```
#ifndef __NETWORKIF_ESP8266_H__
#define __NETWORKIF_ESP8266_H__

//Preferred access point
#define DEF_JOIN_AP_SSID    "NT_ZY"
#define DEF_JOIN_AP_KEY     "12345678"


//ESP8266 network interface task
void NetworkIF_ESP8266_Task(void *pvParameter);

#endif
```

連接到 MQTT 代理人:

```
/** Make a connection to MQTT server in non-blocking mode */
static void example_do_connect(esp_mqtt_client_p client)
{
    if (client == NULL)
    {
        return;
    }

    /*
     * Start a simple connection to open source
     * MQTT server on mosquitto.org
     */
    retries++;
    esp_timeout_remove(mqtt_timeout_cb);
    esp_mqtt_client_connect(mqtt_client, "test.mosquitto.org", 1883, mqtt_cb,
&mqtt_client_info);
}
```

發佈訊息:

```
/*
 * \brief           Timeout callback for MQTT events
 * \param[in]       arg: User argument
 */
```

```
void mqtt_timeout_cb(void *arg)
{
    esp_mqtt_client_p client = arg;
    espr_t res;

    char szPin0Name[20];
    char szPin1Name[20];
    static char tx_data[60];

    GenPinNameStr(szPin0Name, 20, READ_PIN0_PORT, READ_PIN0_PIN);
    GenPinNameStr(szPin1Name, 20, READ_PIN1_PORT, READ_PIN1_PIN);

    if (esp_mqtt_client_is_connected(client))
    {
        sprintf(tx_data, "[%s]: %u, [%s]: %u", szPin0Name, READ_PIN0, szPin1Name,
READ_PIN1);

        if ((res = esp_mqtt_client_publish(client, DEF_TOPIC_STR, tx_data,
ESP_U16(strlen(tx_data)), ESP_MQTT_QOS_EXACTLY_ONCE, 0, tx_data)) == espOK)
        {
            printf("Publishing message ...\r\n");
        }
        else
        {
            printf("Cannot publish...: %d\r\n", (int)res);
        }
    }

    esp_timeout_add(10000, mqtt_timeout_cb, client);
}
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - M480 Series BSP CMSIS V3.04.000
  - IDE 版本
    - Keil uVersion 5.26
- **硬體環境**
  - 電路元件
    - NuMaker-IoT-M487
    - ESP8266 模組
  - 示意圖

M480 通過 UART1 接口與 ESP8266 模組連接。

M480		ESP8266
VCC	↔	VCC
PH.3	→	RESET
UART1_TXD(PH.8)	→	RXD
UART1_RXD(PH.9)	←	TXD
UART1_RTS(PB.8)	→	CTS
UART1_CTS(PB.9)	←	RTS
GND	↔	GND

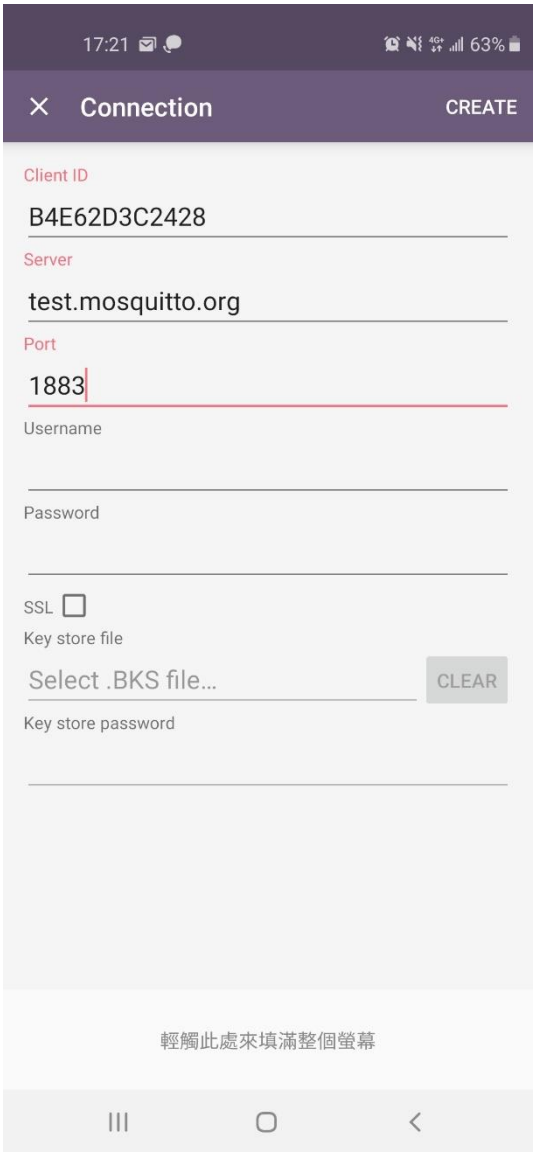
4 目錄資訊

- EC_M480_WiFi_MQTTClient_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code
  - ThirdParty
    - FreeRTOS — A real time operating system available for free download. Its official website is: http://www.freertos.org/
    - ESP_AT_Lib — ESP_AT Library commands parser is generic, platform independent, library for communication with ESP8266 WiFi module using AT commands. Its official website is https://majerle.eu/documentation/esp_at/html/index.html

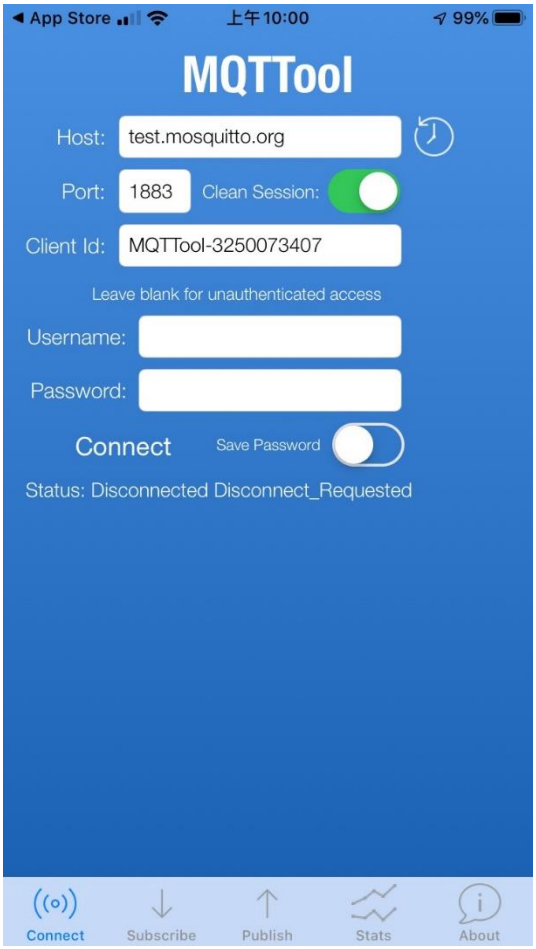
5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 M480_WiFi_MQTTClient.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼
4. 從 Google Play 上安裝 MQTT Dashboard 或者從 App Store 安裝 MQTTool

5. 使用 MQTT Dashboard 或者 MQTTool 建立 MQTT 連線。Client ID 可以是任意的字串。

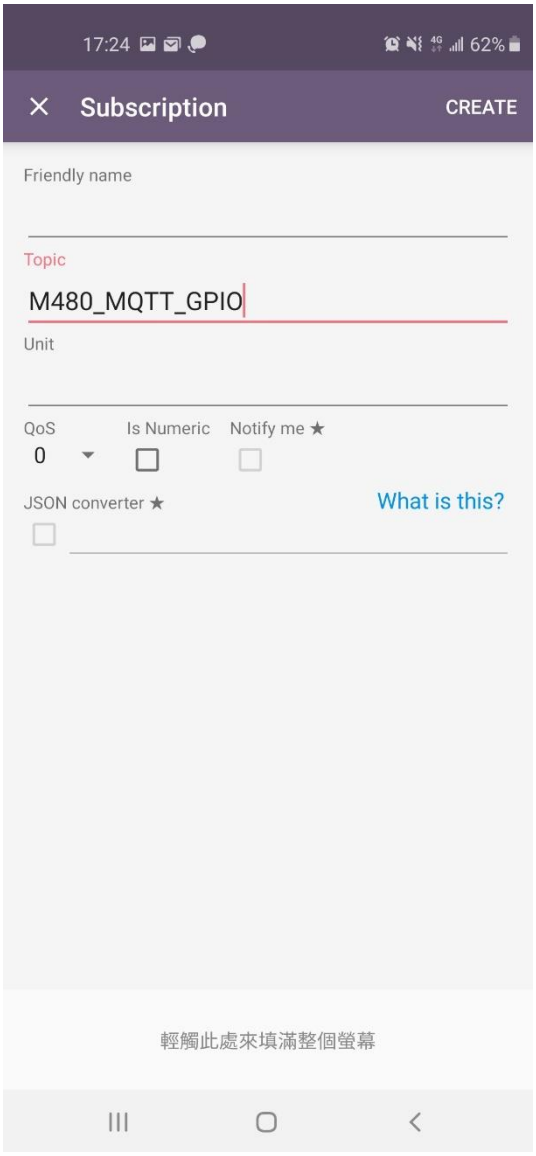


MQTT Dashboard

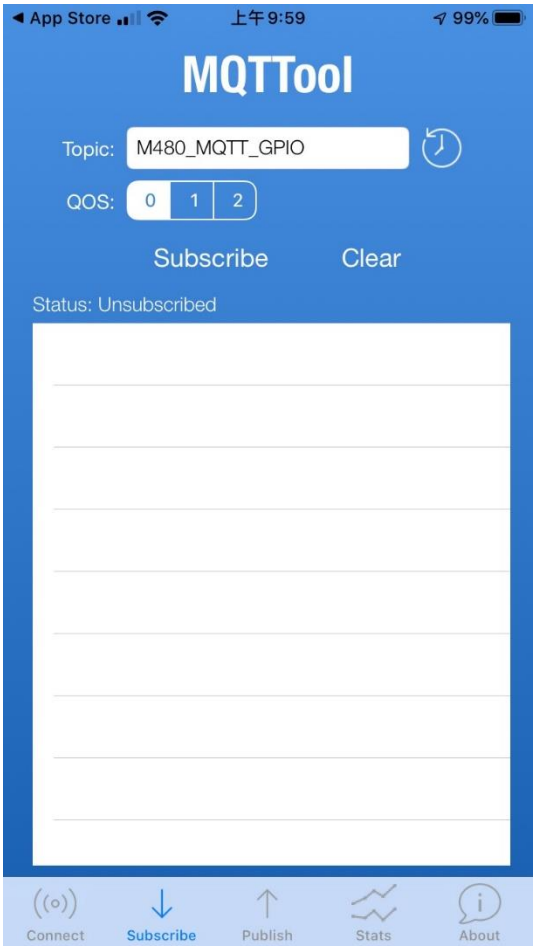


MQTTool

6. 訂閱主題

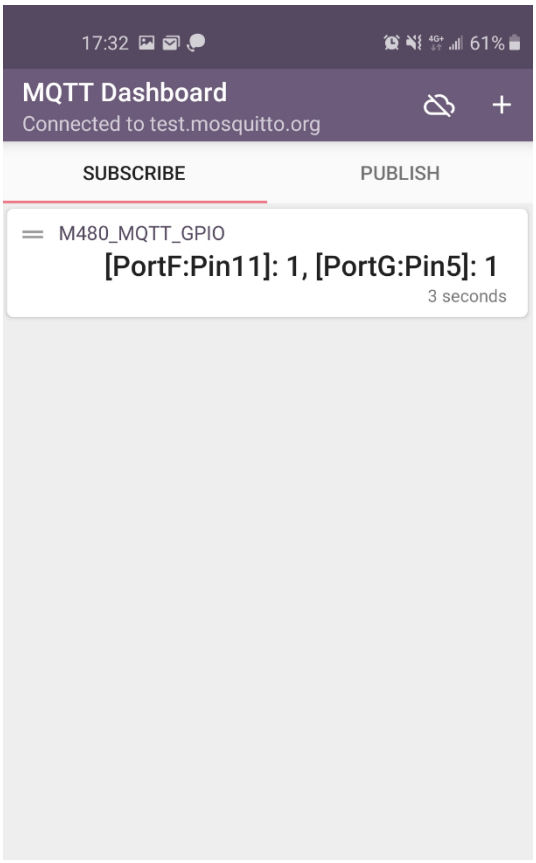


MQTT Dashboard

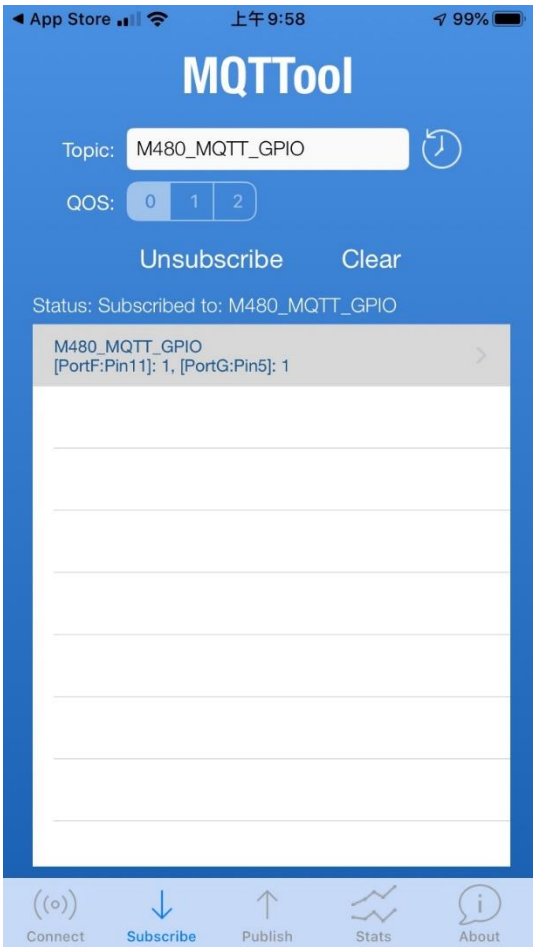


MQTTool

7. 接收訊息



MQTT Dashboard



MQTTool

6 修訂紀錄

Date	Revision	Description
Oct. 8, 2019	1.00	1. 初始發布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.