M480 WiFi MQTT Client

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example uses M480 UART to drive ESP8266 for MQTT client functionality
BSP Version	M480 Series BSP CMSIS V3.04.000
Hardware	NuMaker-IoT-M487 v1.3

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

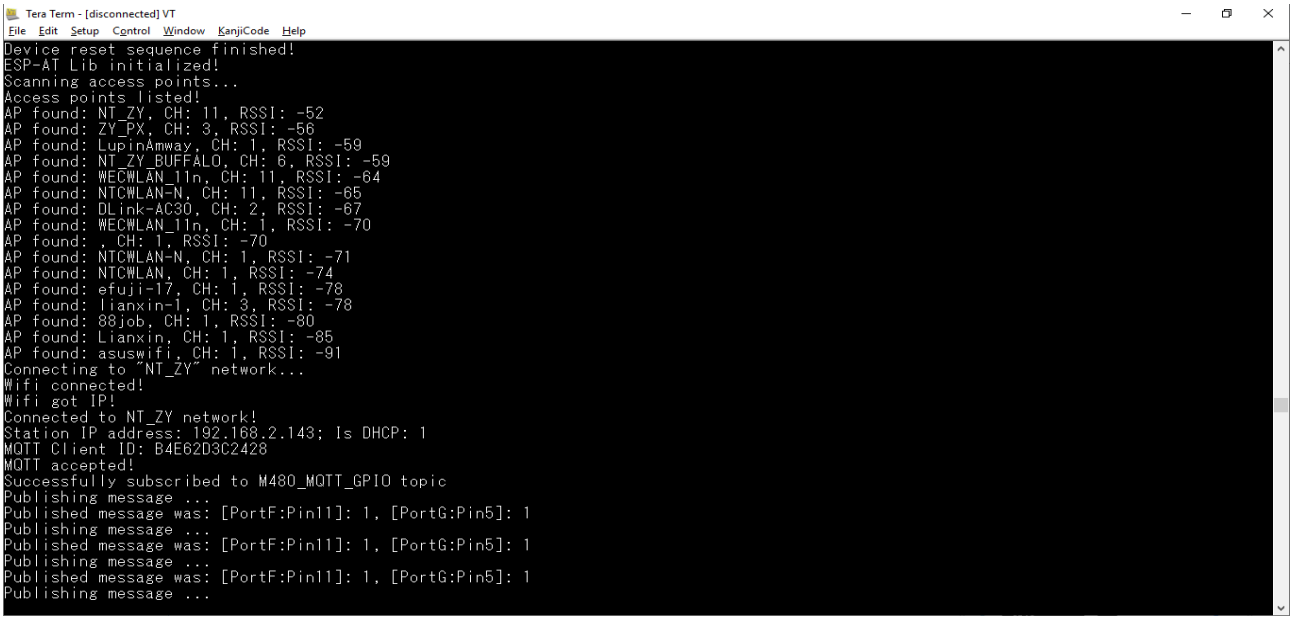
1.1 Introduction

This example code uses M480 UART1 to drive the ESP8266 WiFi module, establishing the TCP/HTTP server capability. When the ESP8266 module is connected to the wireless access point, it establishes a connection with MQTT broker, and publishes the message of NuMaker-IoT-M487 SW2 (GPIO F11) and SW3 (GPIO G5) status.

1.2 Principle

The example code is based on ESP-AT Lib library. ESP-AT Lib is an AT command parser for communication with ESP8266 module. It provided station and access point wireless network management APIs and BSD socket-like APIs. Make it easy for application to build network application with client-server architecture.
MQTT is a messaging protocol based on “Publisher/Subscriber” mechanism. A MQTT system consists of a server and communication clients. The server is often referred to as “Broker”. The client can be the publisher of the message or the subscriber.
The sample code shows the publisher of the message. A message with the topic “M480_MQTT_GPIO” will be published to the broker at every fixed period. Finally, install MQTT Dashboard (for Android) or MQTTool (for iOS) as a subscriber on to receive the message with the topic “M480_MQTT_GPIO”.

1.3 Demo Result



2 Code Description

Setup preferred access point：

```
#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
```

Connect to MQTT broker：

```
/** 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);
}
```

Publish message：

```
/*
 * \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 Software and Hardware Environment

Software Environment

- BSP version
  - M480 Series BSP CMSIS V3.04.000
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuMaker-IoT-M487
  - ESP8266 module
- Diagram

The M480 is connected to the ESP8266 module via UART1 interface.

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

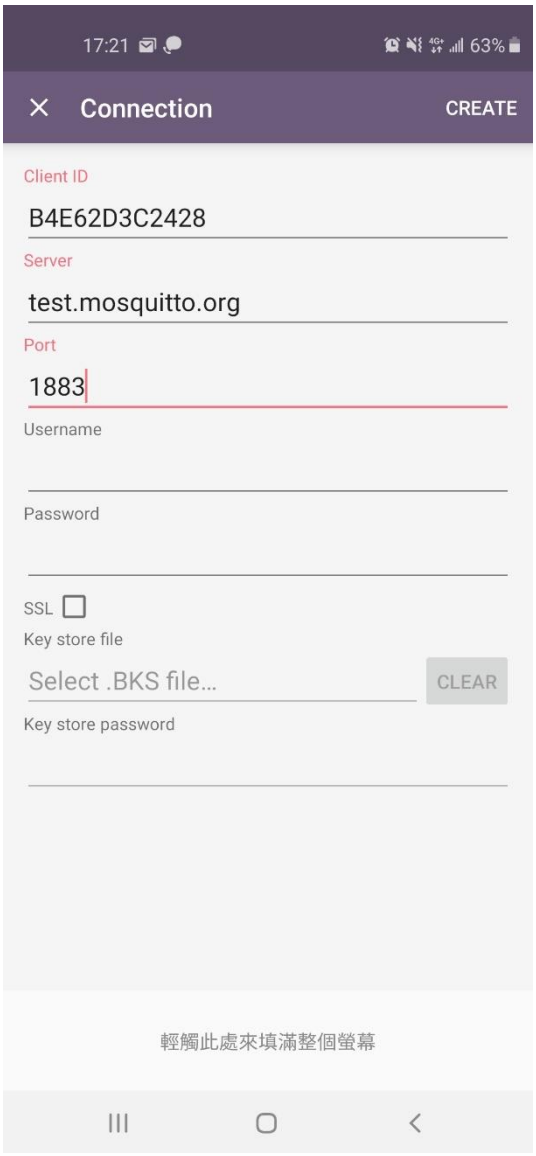
4 Directory Information

- 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

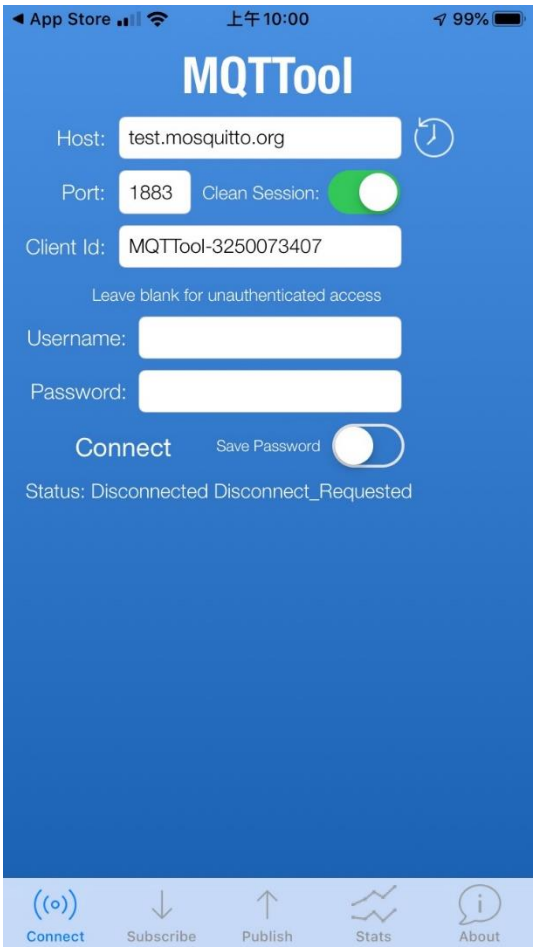
5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click M480_WiFi_MQTTClient.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run
4. Install MQTT Dashboard from Google Play or MQTTool from App Store

5. Using MQTT Dashboard or MQTTool to create MQTT connection. The Client ID string can be arbitrary.

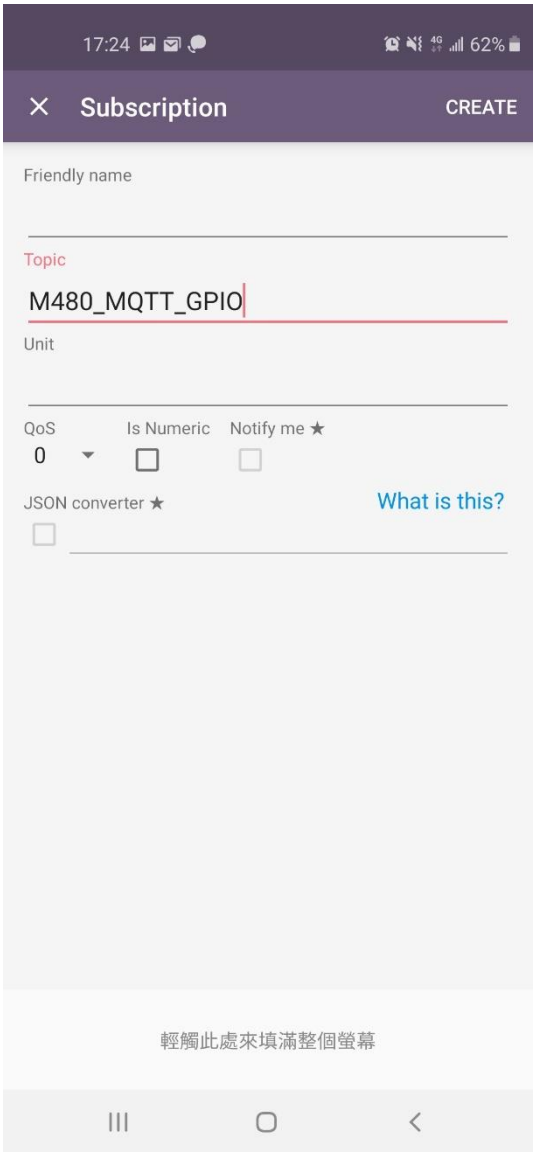


MQTT Dashboard

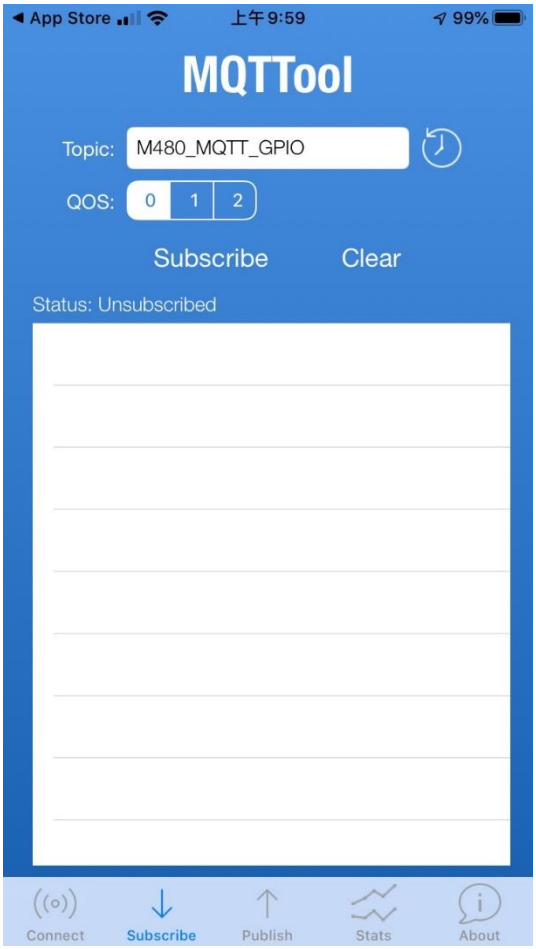


MQTTool

6. Subscribe topic

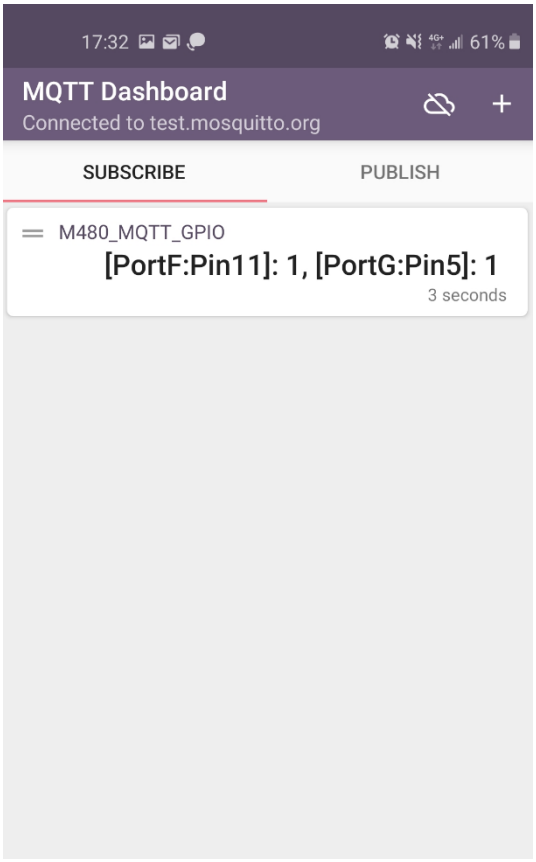


MQTT Dashboard

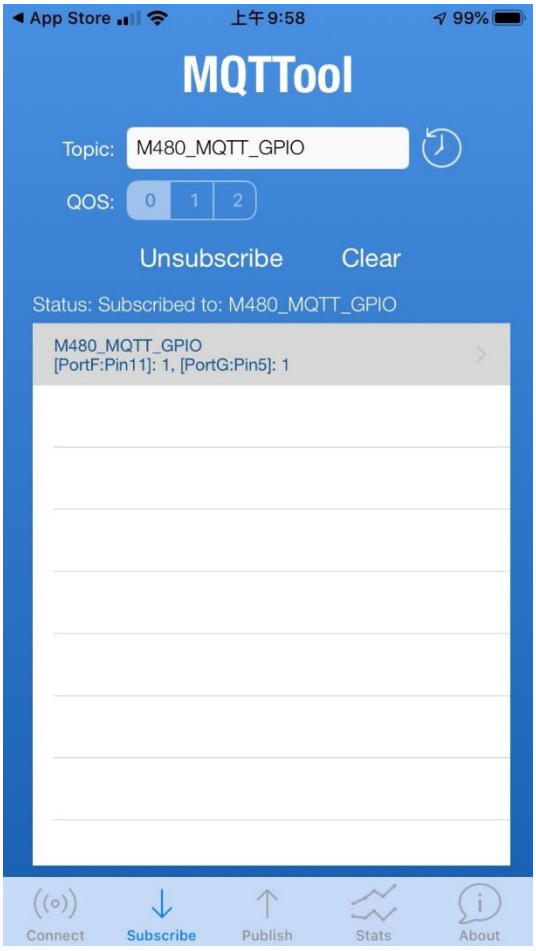


MQTTool

7. Receive message



MQTT Dashboard



MQTTool

6 Revision History

Date	Revision	Description
Oct. 9, 2019	1.00	1. Initially issued.

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.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.