M480 WiFi MQTT 客户端

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	本范例使用 M480 UART 驱动 ESP8266，实现 MQTT 客户端功能
BSP 版本	M480 Series BSP CMSIS V3.04.000
开发平台	NuMaker-IoT-M487 v1.3

1 功能介绍

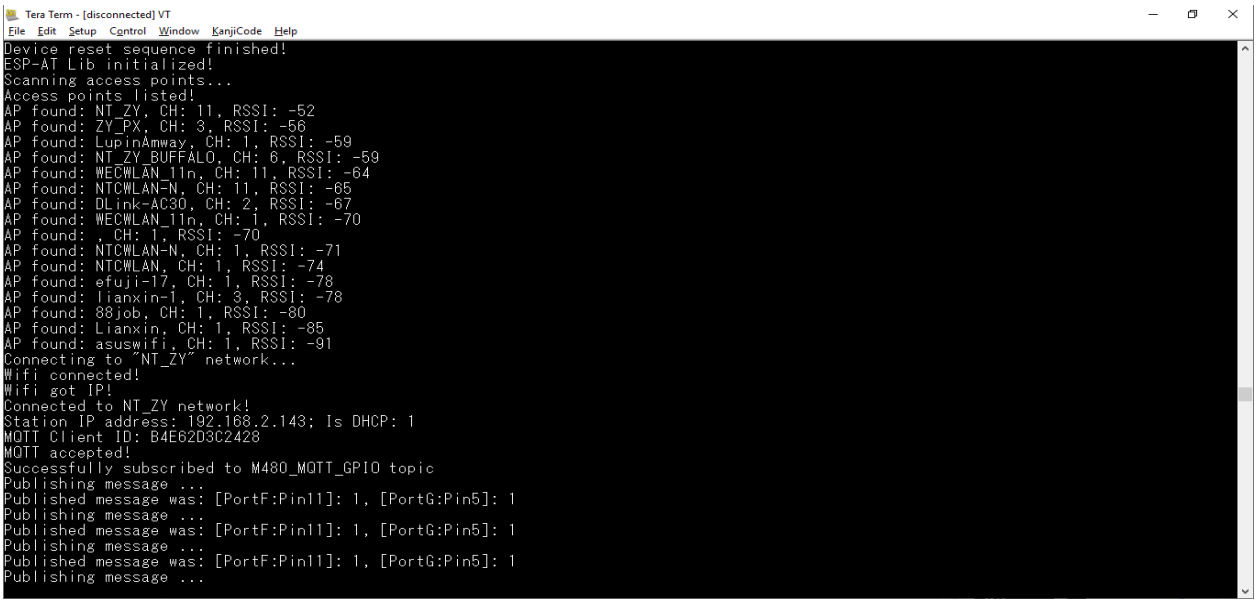
1.1 简介

本范例程序代码使用M480系列芯片，通过UART1驱动ESP8266 WiFi模块，实现MQTT客户端功能。当ESP8266模块连接上无线存取点后，开始建立与MQTT代理人的联机，并且每10秒发布NuMaker-IoT-M487板子上SW2(GPIO F11)和SW3(GPIO G5)按键状态的讯息。

1.2 原理

范例程序核心使用了ESP-AT Lib函式库。ESP-AT Lib是一个AT命令解析器，用于使用AT命令与ESP8266 WiFi模块通信。ESP-AT Lib提供了Station和Access point无线网络管理API，并提供了BSD socket相似的API，方便应用程序建立主从式架构的网络应用。
MQTT是一种基于”发布(Publisher)/订阅(Subscriber)”机制的讯息传输协议。一个MQTT系统由服务器和通信的客户端所组成，而服务器通常称为”代理人(Broker)”。客户端可以是讯息的发布者，也可以是订阅者。
在范例程序里，展示的是客户端讯息的发布者，固定会发布主题为”M480_MQTT_GPIO”的讯息到代理人端。代理人端使用的是”test.mosquitto.org”服务器。最后在Android手机上使用MQTT Dashboard或者在iOS装置上使用MQTTool在做为订阅者，来接收主题为”M480_MQTT_GPIO”的讯息。

1.3 执行结果



2 代码介绍

设置连接无线存取点的 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
```

连接到 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);
}
```

发布讯息:

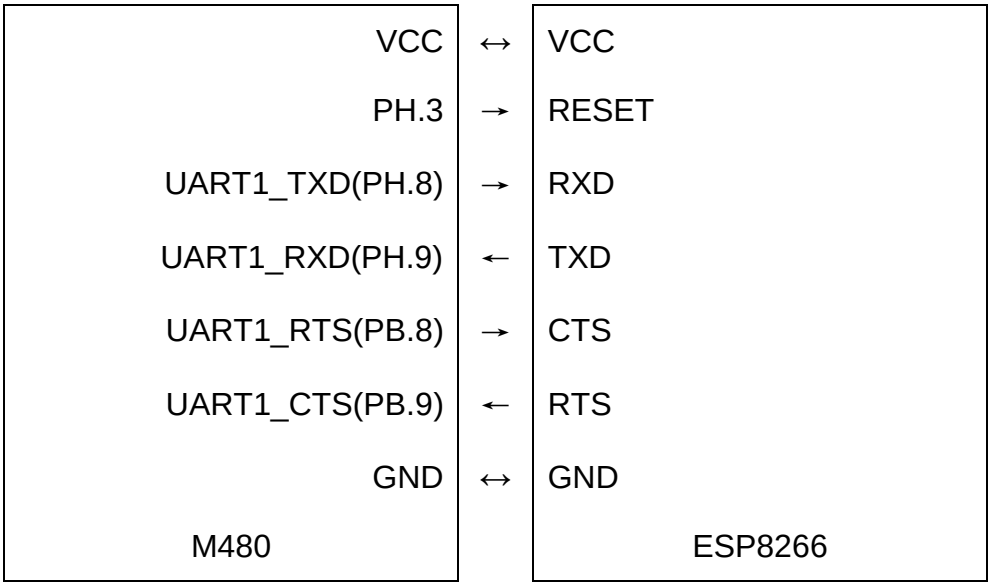
```
/*
 * \brief           Timeout callback for MQTT events
 * \param[in]       arg: User argument
 */
```

```
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 软件与硬件环境

- 软件环境
  - BSP 版本
    - M480 Series BSP CMSIS V3.04.000
  - IDE 版本
    - Keil uVersion 5.26
- 硬件环境
  - 电路组件
    - NuMaker-IoT-M487
    - ESP8266 模块
  - 示意图

M480 通过 UART1 接口与 ESP8266 模块连接。



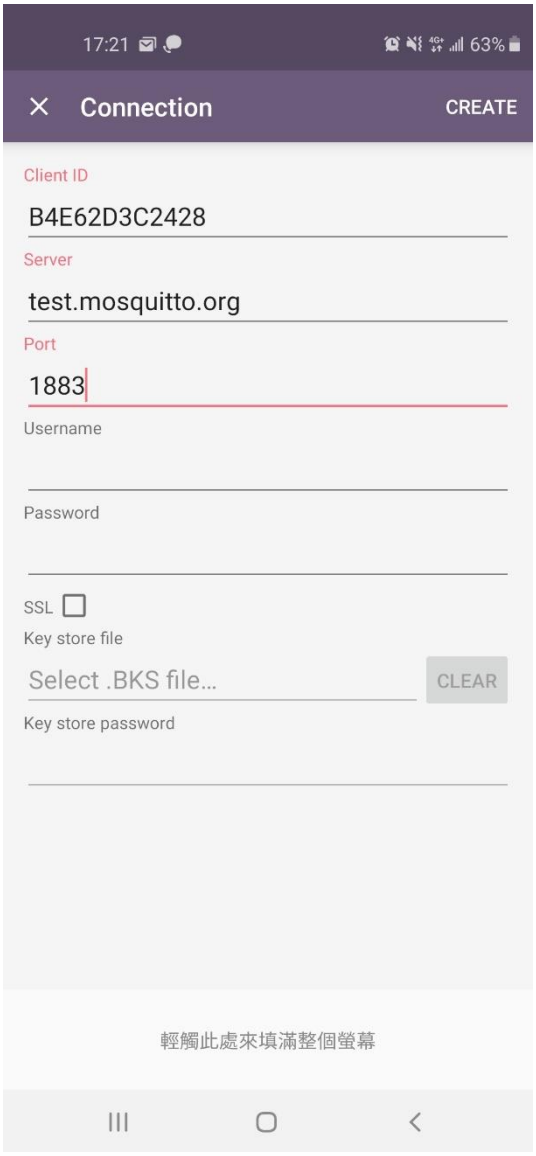
4 目录信息

- 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

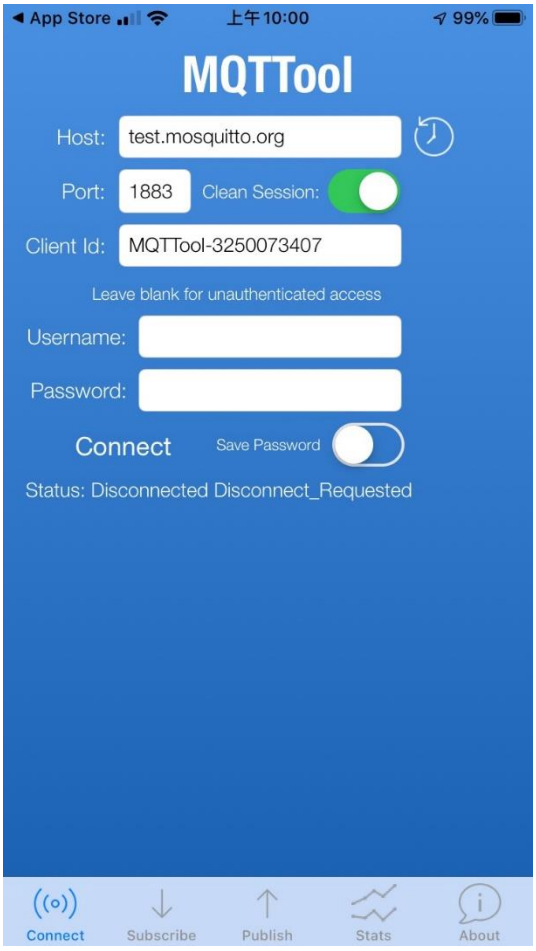
5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 M480_WiFi_MQTTClient.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码
4. 从 Google Play 上安装 MQTT Dashboard 或者从 App Store 安装 MQTTool

5. 使用 MQTT Dashboard 或者 MQTTool 建立 MQTT 联机。Client ID 可以是任意的字符串。

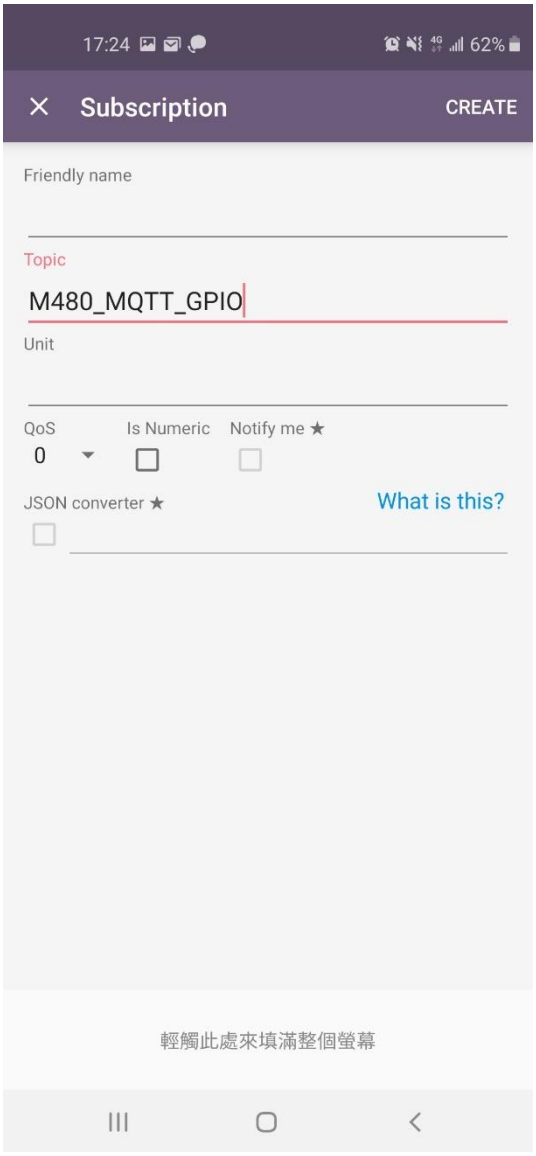


MQTT Dashboard

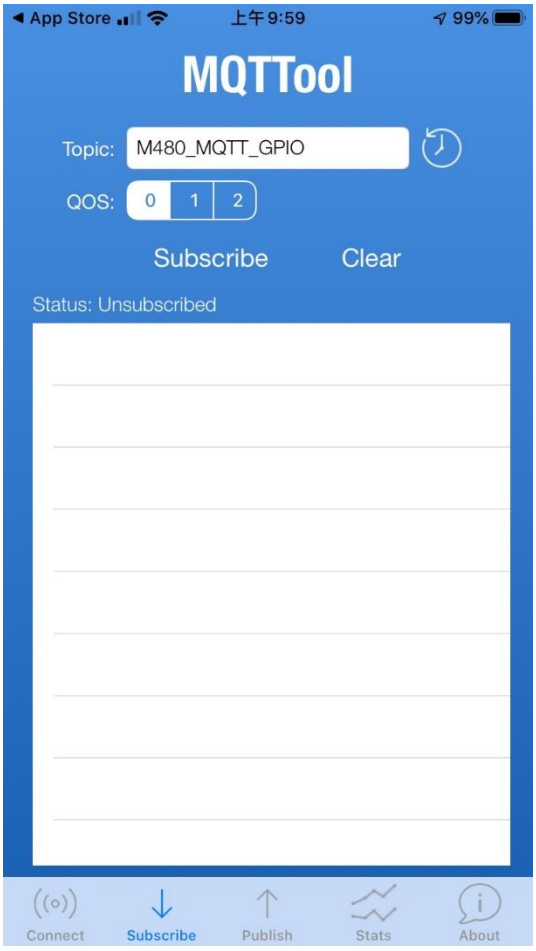


MQTTool

6. 订阅主题

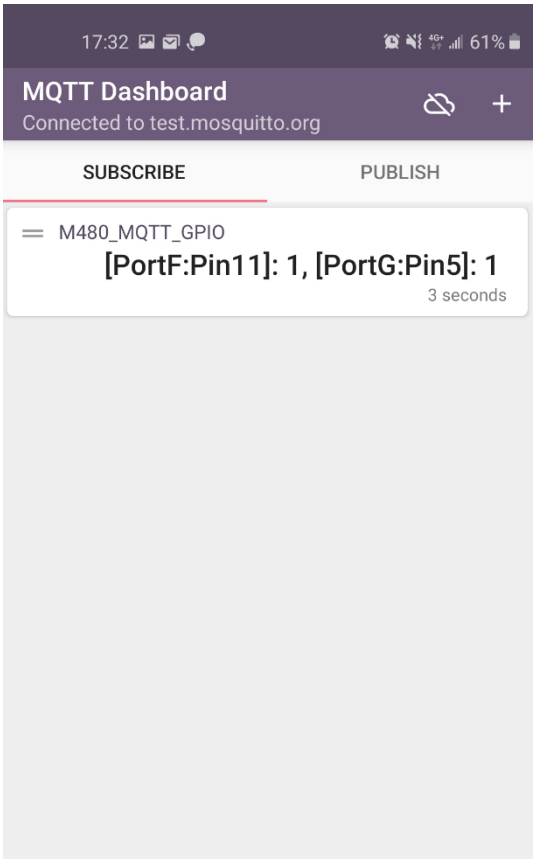


MQTT Dashboard

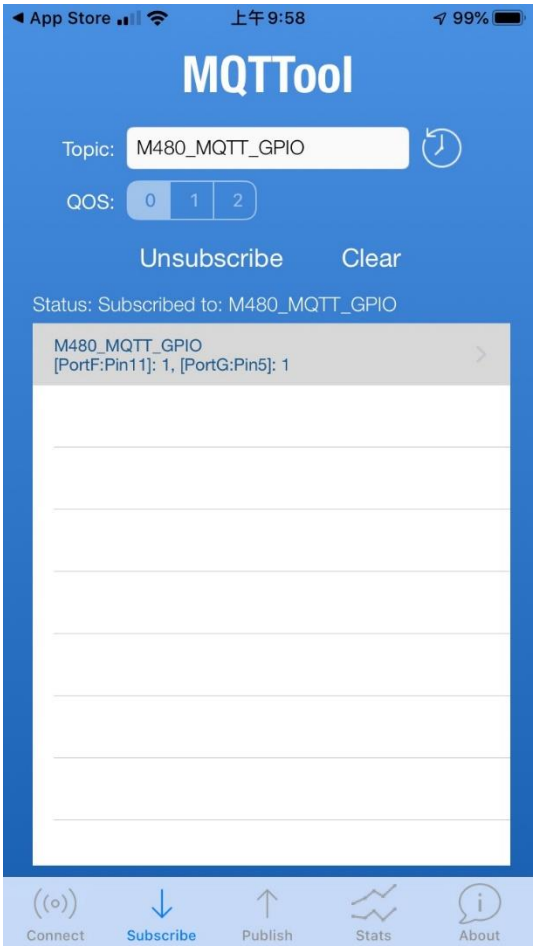


MQTTool

7. 接收讯息



MQTT Dashboard



MQTTool

6 修订纪录

Date	Revision	Description
Oct. 8, 2019	1.00	1. 初始发布.

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.