M480 WiFi TCP Server

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example uses M480 UART to drive ESP8266 for TCP server 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

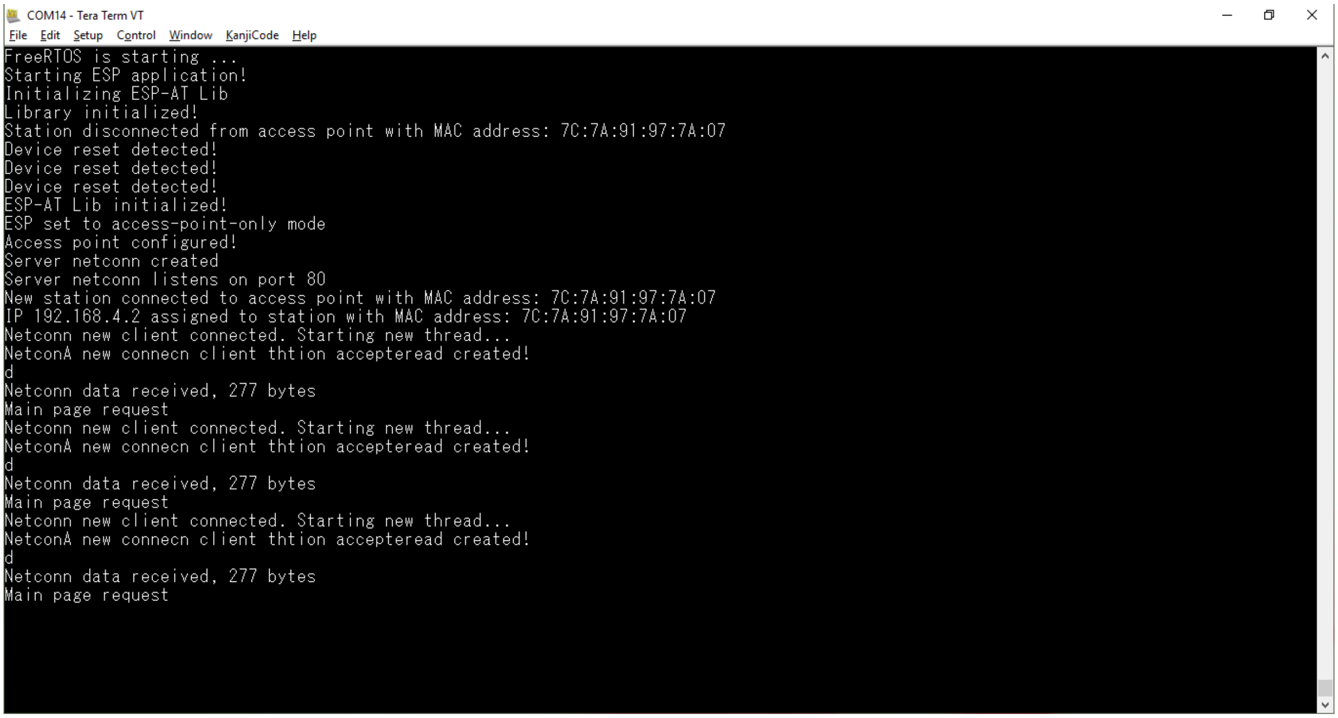
This example code uses M480 UART1 to drive the ESP8266 WiFi module, create wireless access point and provide TCP server capability. When the computer is connected to ESP8266 access point via WiFi, you can turn on the browser to observe the status of the NuMaker-IoT-M487 SW2 (GPIO F11) and SW3 (GPIO G5) buttons.

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.

The program puts the ESP8266 module into access point mode, and then turns on HTTP 80 port waiting for the network connection. User can use computer to join the wireless access point with the SSID “ESP_AccessPoint” and the key “12345678”. Finally, open browser to observe the status of the NuMaker-IoT-M487 SW2 and SW3 buttons.

1.3 Demo Result



```
FreeRTOS is starting ...
Starting ESP application!
Initializing ESP-AT Lib
Library initialized!
Station disconnected from access point with MAC address: 7C:7A:91:97:7A:07
Device reset detected!
Device reset detected!
Device reset detected!
ESP-AT Lib initialized!
ESP set to access-point-only mode
Access point configured!
Server netconn created
Server netconn listens on port 80
New station connected to access point with MAC address: 7C:7A:91:97:7A:07
IP 192.168.4.2 assigned to station with MAC address: 7C:7A:91:97:7A:07
Netconn new client connected. Starting new thread...
NetconA new connecn client thtion acceptered created!
d
Netconn data received, 277 bytes
Main page request
Netconn new client connected. Starting new thread...
NetconA new connecn client thtion acceptered created!
d
Netconn data received, 277 bytes
Main page request
Netconn new client connected. Starting new thread...
NetconA new connecn client thtion acceptered created!
d
Netconn data received, 277 bytes
Main page request
```

2 Code Description

Set the ESP8266 module to Access point mode：

```
    /* Enable access point only mode */
    if ((res = esp_set_wifi_mode(ESP_MODE_AP, 1, NULL, NULL, 1)) == espOK)
    {
        printf("ESP set to access-point-only mode\r\n");
    }
    else
    {
        printf("Problems setting ESP to access-point-only mode: %d\r\n", (int)res);
    }

    /* Configure access point */
    res = esp_ap_configure("ESP_AccessPoint", "12345678", 10, ESP_ECN_WPA2_PSK, 5, 0, 0,
NULL, NULL, 1);

    if (res == espOK)
    {
        printf("Access point configured!\r\n");
    }
```

Enable 80 port service：

```
    /*
     * First create a new instance of netconn
     * connection and initialize system message boxes
     * to accept clients and packet buffers
     */
    server = esp_netconn_new(ESP_NETCONN_TYPE_TCP);

    if (server != NULL)
    {
        printf("Server netconn created\r\n");

        /* Bind network connection to port 80 */
        res = esp_netconn_bind(server, 80);
        if (res == espOK)
        {

            printf("Server netconn listens on port 80\r\n");
            /*
             * Start listening for incoming connections
             * on previously binded port
             */
            res = esp_netconn_listen(server);

            while (1)
            {
                /*
                 * Wait and accept new client connection
                 *
                 * Function will block thread until
                 * new client is connected to server
                 */
                res = esp_netconn_accept(server, &client);

                if (res == espOK)
                {
                    printf("Netconn new client connected. Starting new thread...\r\n");

                    /*
                     * Start new thread for this request.
                     *
                     * Read and write back data to user in separated thread
                     * to allow processing of multiple requests at the same time
                     */
                    if (esp_sys_thread_create(NULL, "client",
(esp_sys_thread_fn)netconn_server_processing_thread, client, 512, ESP_SYS_THREAD_PRIO))
                    {
                        printf("Netconn client thread created\r\n");
                    }
                    else
                    {
                        printf("Netconn client thread creation failed!\r\n");
                        /* Force close & delete */
                        esp_netconn_close(client);
                        esp_netconn_delete(client);
                    }
                }
                else
                {
                    printf("Netconn connection accept error!\r\n");
                    break;
                }
            }
        }
        else
        {
            printf("Netconn server cannot bind to port\r\n");
        }
    }
    else
    {
        printf("Cannot create server netconn\r\n");
    }
```

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.

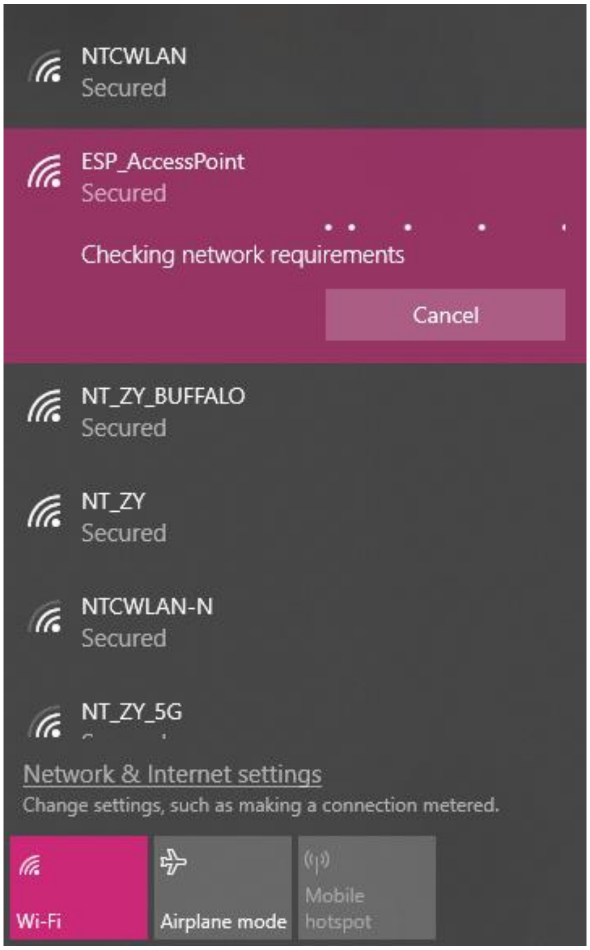
VCC	↔	VCC
PH.3	→	RESET
UART1_TXD(PH.8)	→	RXD
UART1_RXD(PH.9)	←	TXD
UART1_RTS(PB.8)	→	CTS
UART1_CTS(PB.9)	←	RTS
GND	↔	GND
M480		ESP8266

4 Directory Information

🗁 EC_M480_WiFi_TCPServer_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

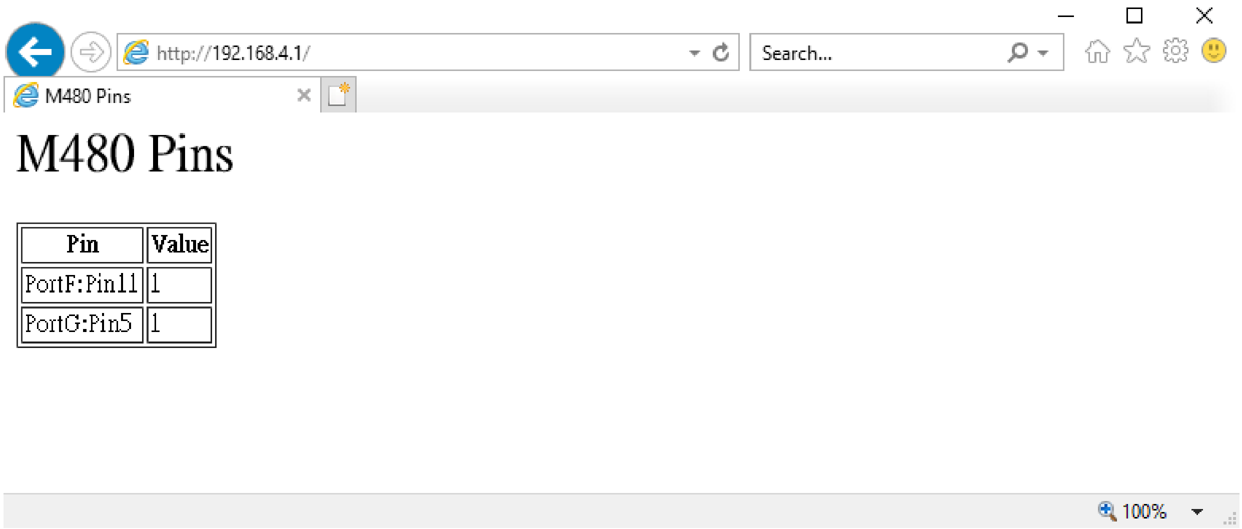
1. Browsing into sample code folder by Directory Information (section 4) and double click M480_WiFi_TCPServer.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run
4. Computer connect to “ESP_AccessPoint” WiFi access point (Key:12345678)



Checking network requirements

Change settings, such as making a connection metered.

5. Open browser and enter http://192.168.4.1



M480 Pins

Pin	Value
PortF:Pin11	1
PortG:Pin5	1

6 Revision History

Date	Revision	Description
Oct. 8, 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.