Use NUC240 UART to Drive ESP8266

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses NUC240 UART to drive ESP8266
BSP Version	NUC230_240_Series_BSP_CMSIS_V3.01.001
Hardware	NuTiny-SDK-NUC240V

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

ESP-01S is a small-sized ESP8266-based module produced by Espressif.
The ESP-01S ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP-01S ESP8266 module comes pre-programmed with an AT command set firmware.
The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating via an IP network. Major internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP. Applications that do not require reliable data stream service may use the User Datagram Protocol (UDP), which provides a connectionless datagram service that emphasizes reduced latency over reliability.
This sample code uses NUC240 series MCU to drive ESP-01S Module through UART0 interface, using the built-in TCP protocol stack of esp8266 to realize the function of TCP Server. When the Computer connects ESP-01S module through Wi-Fi, users can use the serial port of computer tools and network assistant tools to view the execution results.

With ESP8266 configured as a TCP server, multiple connections should be allowed，since there may be more than one client connecting to the ESP8266.
Here shows how TCP server is realized when ESP8266 is working in softAP mode.

1. Set Wi-Fi mode.
   AT+CWMODE=3 // softAP+station mode
   Response:
   OK
2. Enable multiple connections.
   AT+CIPMUX=1
   Response:
   OK
3. Create a Soft AP and not no password.

AT+CWSAP_CUR=Nuvoton_AP,,5,0,4,0
Response:
OK

4. Create a TCP server.
   AT+CIPSERVER=1,8080 // port = 8080
   Response:
   OK
5. Connect PC to ESP8266 softAP. Then using a network tool on PC to create a TCP client and connect to the TCP server that ESP8266 created.
6. When ESP8266 received data from server, it will prompt message below:
   +IPD,0,n:xxxxxxxxxx // received n bytes, data=xxxxxxxxxx

2 Code Description

When key pressed, ESP8266 is configured as AP mode and runs as TCP server:

```
void Key_Process(uint8_t u8Key)
{
    uint32_t u32cnt;

    if (u8Key == KEY_NONE)
    {
        return;
    }
    switch (u8Key)
    {
        case KEY_SW1:
            // Reset ESP8266
            UART_Write(UART0, atCmdRst.pu8string, atCmdRst.u8CharNum);
            break;
        case KEY_SW2:
            // Set ESP8266 STA+AP mode
            UART_Write(UART0, atCmdCwmode3.pu8string, atCmdCwmode3.u8CharNum);
            for (u32cnt = 0; u32cnt < 0xffff; u32cnt++);
            // Set ESP8266 Multi-connection
            UART_Write(UART0, atCmdCipmux1.pu8string, atCmdCipmux1.u8CharNum);
            for (u32cnt = 0; u32cnt < 0xffff; u32cnt++);
            // Set ESP8266 AP Name, No Password
            UART_Write(UART0, atCmdAp.pu8string, atCmdAp.u8CharNum);
            for (u32cnt = 0; u32cnt < 0xffff; u32cnt++);
            // Set ESP8266 TCP Server mode, Port number 8080
            UART_Write(UART0, atCmdCipserver1.pu8string, atCmdCipserver1.u8CharNum);
            for (u32cnt = 0; u32cnt < 0xffff; u32cnt++);

            u8ReceiveStringArrayHeader = u8ReceiveStringArrayTail = 0;
            memset(ReceiveArray, 0, sizeof(ReceiveArray));
        default:
            break;
    }
}
```

3 Software and Hardware Environment

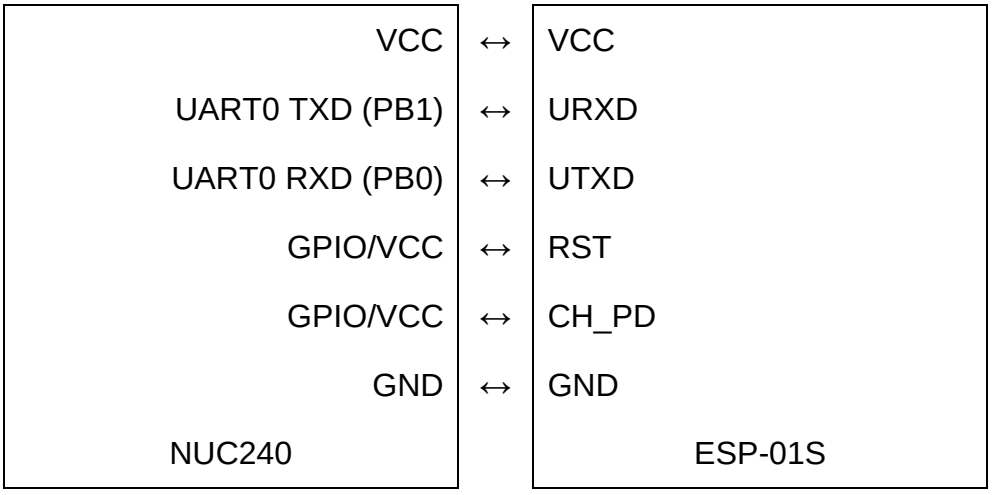
Software Environment

- BSP version
  - NUC230_240_Series_BSP_CMSIS_V3.01.001
- IDE version
  - Keil uVersion 5.29

Hardware Environment

- Circuit components
  - NuTiny-SDK-NUC240V
  - NuEdu-Basic01 V2.0
  - ESP-01S Module
- Diagram

NUC240 connects with ESP-01S module using UART0 interface. Use UART1 to print message.



4 Directory Information

- EC_NUC240_UART_ESP8266_V1.00
  - Document — 4a-esp8266_at_instruction_set_en.pdf
  - 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

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click UART_ESP8266.uvprojx.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Sept. 17, 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.