M480_FreeRTOS_LwIP_Http_Server

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This code is an http server that is implemented with LwIP under FreeRTOS to control a LED on evaluation board.
BSP Version	M480 Series BSP CMSIS V3.04.000
Hardware	NuMaker-PFM-M487

1 Function Description

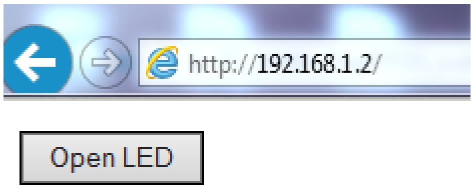
1.1 Introduction

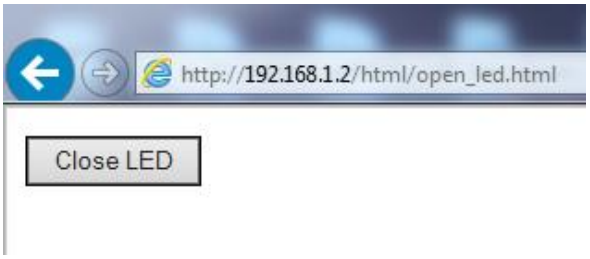
This code is an http server that is implemented with LwIP under FreeRTOS. This FreeRTOS system creates a task that control with a LED on evaluation board.

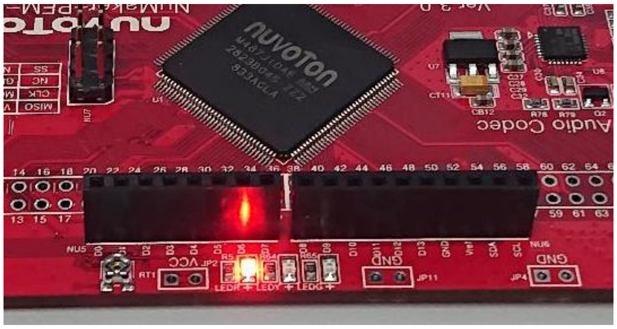
1.2 Principle

This FreeRTOS sample code creates a task. This task acts as an http server, which is implemented with LwIP. It is configured statically to IP address 192.168.1.2:80. The PC could use a web browser to connect with this http server by the webpage http://192.168.1.2/. When user clicks the open LED button, the LED on evaluation board will be turn on. Then, the webpage will change to a close LED page. User also could click the close LED button to turn off the LED. And the webpage will back to the open LED page.

1.3 Demo Result







2 Code Description

Set LwIP network interface and create task for http server：

```
static void vWebTask(void *pvParameters)
{
    ip_addr_t ipaddr;
    ip_addr_t netmask;
    ip_addr_t gw;
    /*IP address setting*/
    IP4_ADDR(&gw, 192,168,1,1);
    IP4_ADDR(&ipaddr, 192,168,1,2);
    IP4_ADDR(&netmask, 255,255,255,0);
    printf("Local IP:192.168.1.2\n");
    /*network interface initial*/
    tcpip_init(NULL, NULL);
    netif_add(&netif, &ipaddr, &netmask, &gw, NULL, ethernetif_init, tcpip_input);
    netif_set_default(&netif);
    netif_set_up(&netif);

    /*EMAC interrupt enable*/
    NVIC_SetPriority(EMAC_TX_IRQn, configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY + 1);
    NVIC_EnableIRQ(EMAC_TX_IRQn);
    NVIC_SetPriority(EMAC_RX_IRQn, configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY + 1);
    NVIC_EnableIRQ(EMAC_RX_IRQn);

    http_server_netconn_init();  /*Create http_server task*/
    vTaskSuspend(NULL);
}
```

Http server task function：

```
static void http_server_netconn_thread(void *arg)
{
    struct netconn *conn, *newconn;
    err_t err;

    /* Create a new TCP connection handle */
    conn = netconn_new(NETCONN_TCP);

    if(conn!= NULL) {
        /* Bind to port 80 (HTTP) with default IP address */
        err = netconn_bind(conn, NULL, 80);
        if(err == ERR_OK) {
            /* Put the connection into LISTEN state */
            netconn_listen(conn);
            while(1) {
                /* accept any incoming connection */
                netconn_accept(conn,&newconn);
                if(newconn) {
                    /* serve connection */
                    http_server_serve(newconn);

                    /* delete connection */
                    netconn_delete(newconn);
                }
            }
        } else {
            printf("can not bind netconn");
        }
    } else {
        printf("can not create netconn");
    }
    while(1);
}

/*Webpage data*/
static const char* close_led = "<html><body><form name='input' action='/html/open_led.html' method='get'><input type='submit' value='Open LED' /></form></body></html>";
static const char* open_led = "<html><body><form name='input' action='/html/close_led.html' method='get'><input type='submit' value='Close LED' /></form></body></html>";


void http_server_serve(struct netconn *conn)
{
    struct netbuf *inbuf;
    char* buf;
    u16_t buflen;
    struct fs_file * file;

    /* Read the data from the port, blocking if nothing yet there.
     We assume the request is in one netbuf */
    netconn_recv(conn,&inbuf);
    if(inbuf != NULL) {
        if(netconn_err(conn) == ERR_OK) {
            netbuf_data(inbuf, (void**)&buf, &buflen);

            /* Is this an HTTP GET command? (only check the first 5 chars, since
            there are other formats for GET, and we're keeping it very simple )*/
            if((buflen >=5) && (strncmp(buf, "GET /", 5) == 0)) {
                /* Check if request to get html data */
                if(strncmp((char const *)buf,"GET /html/open_led.html", strlen("GET
                           /html/open_led.html")) == 0) {
                    netconn_write(conn, (const unsigned char*)open_led,
                                  (size_t)strlen(open_led), NETCONN_NOCOPY);
                    PH0 = 0;  //Open led
                } else if(strncmp(buf, "GET /html/close_led.html", strlen("GET
                                  /html/close_led.html")) == 0) {
                    netconn_write(conn, (const unsigned char*)close_led,
                                  (size_t)strlen(close_led), NETCONN_NOCOPY);
                    PH0 = 1;  //Close led
                } else if((strncmp(buf, "GET /index.html", 15) == 0)||(strncmp(buf, "GET /
                                   ", 6) == 0)) {
                    netconn_write(conn, (const unsigned char*)close_led,
                                  (size_t)strlen(close_led), NETCONN_NOCOPY);
                    PH0 = 1;  //Close led
                } else {
                    /* Load Error page */
                    file = fs_open("/404.html");
                    netconn_write(conn, (const unsigned char*)(file->data), (size_t)file-
                                  >len, NETCONN_NOCOPY);
                    fs_close(file);
                }
            }
        }
    }
    /* Close the connection (server closes in HTTP) */
    netconn_close(conn);

    /* Delete the buffer (netconn_recv gives us ownership,
     so we have to make sure to deallocate the buffer) */
    netbuf_delete(inbuf);
}
```

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-PFM-M487 v3.0
- Diagram

M480		Ethernet jack (CN1)
PA.6	←	RMII_RXERR
PA.7	←	RMII_CRSDV
PC.6	←	RMII_RXD1
PC.7	←	RMII_RXD0
PC.8	←	RMII_REFCLK
PE.8	→	RMII_MDC
PE.9	↔	RMII_MDIO
PE.10	→	RMII_TXD0
PE.11	→	RMII_TXD1
PE.12	→	RMII_TXEN

4 Directory Information

🗁 EC_M480_FreeRTOS_LwIP_Http_Server_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/
		🗁 lwIP	A widely used open source TCP/IP stack designed for embedded systems. Its official website is: http://savannah.nongnu.org/projects/lwip/

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click M480_FreeRTOS_LwIP_Http_Server.uvproj.
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
Oct. 11, 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.