Lab 7 Report – Ryan King

Lab 7 Part 1

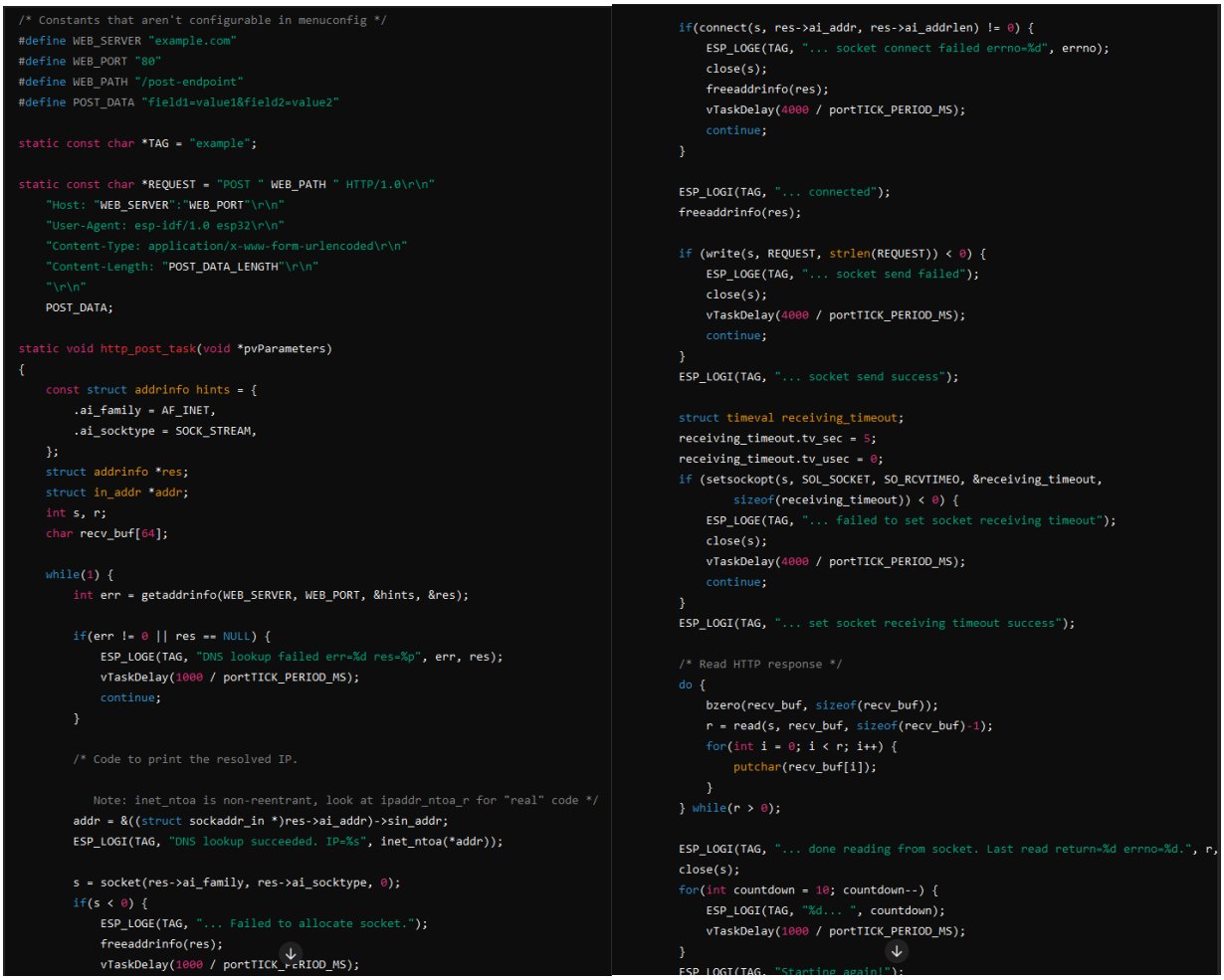
No external resources for part 1, just the lecture code

Lab 7 Part 2

For part 2 I used ChatGPT to ask how to convert from the http_request.c to a program that could handle POST.

https://chatgpt.com/share/5564d6c4-5976-4203-806a-59a69394e0ac

“How would I edit this code in the esp-idf framework for the esp32c3 to make a POST request instead of GET” (plus http_request.c code)



```c
/* Constants that aren't configurable in menuconfig */
#define WEB_SERVER "example.com"
#define WEB_PORT "80"
#define WEB_PATH "/post-endpoint"
#define POST_DATA "field1=value1&field2=value2"

static const char *TAG = "example";

static const char *REQUEST = "POST " WEB_PATH " HTTP/1.0\r\n"
    "Host: "WEB_SERVER":"WEB_PORT"\r\n"
    "User-Agent: esp-idf/1.0 esp32\r\n"
    "Content-Type: application/x-www-form-urlencoded\r\n"
    "Content-Length: "POST_DATA_LENGTH"\r\n"
    "\r\n"
    POST_DATA;

static void http_post_task(void *pvParameters)
{
    const struct addrinfo hints = {
        .ai_family = AF_INET,
        .ai_socktype = SOCK_STREAM,
    };
    struct addrinfo *res;
    struct in_addr *addr;
    int s, r;
    char recv_buf[64];

    while(1) {
        int err = getaddrinfo(WEB_SERVER, WEB_PORT, &hints, &res);

        if(err != 0 || res == NULL) {
            ESP_LOGE(TAG, "DNS lookup failed err=%d res=%p", err, res);
            vTaskDelay(1000 / portTICK_PERIOD_MS);
            continue;
        }

        /* Code to print the resolved IP.

           Note: inet_ntoa is non-reentrant, look at ipaddr_ntoa_r for "real" code */
        addr = &((struct sockaddr_in *)res->ai_addr)->sin_addr;
        ESP_LOGI(TAG, "DNS lookup succeeded. IP=%s", inet_ntoa(*addr));

        s = socket(res->ai_family, res->ai_socktype, 0);
        if(s < 0) {
            ESP_LOGE(TAG, "... Failed to allocate socket.");
            freeaddrinfo(res);
            vTaskDelay(1000 / portTICK_PERIOD_MS);
```

```c
        if(connect(s, res->ai_addr, res->ai_addrlen) != 0) {
            ESP_LOGE(TAG, "... socket connect failed errno=%d", errno);
            close(s);
            freeaddrinfo(res);
            vTaskDelay(4000 / portTICK_PERIOD_MS);
            continue;
        }

        ESP_LOGI(TAG, "... connected");
        freeaddrinfo(res);

        if (write(s, REQUEST, strlen(REQUEST)) < 0) {
            ESP_LOGE(TAG, "... socket send failed");
            close(s);
            vTaskDelay(4000 / portTICK_PERIOD_MS);
            continue;
        }
        ESP_LOGI(TAG, "... socket send success");

        struct timeval receiving_timeout;
        receiving_timeout.tv_sec = 5;
        receiving_timeout.tv_usec = 0;
        if (setsockopt(s, SOL_SOCKET, SO_RCVTIMEO, &receiving_timeout,
                sizeof(receiving_timeout)) < 0) {
            ESP_LOGE(TAG, "... failed to set socket receiving timeout");
            close(s);
            vTaskDelay(4000 / portTICK_PERIOD_MS);
            continue;
        }
        ESP_LOGI(TAG, "... set socket receiving timeout success");

        /* Read HTTP response */
        do {
            bzero(recv_buf, sizeof(recv_buf));
            r = read(s, recv_buf, sizeof(recv_buf)-1);
            for(int i = 0; i < r; i++) {
                putchar(recv_buf[i]);
            }
        } while(r > 0);

        ESP_LOGI(TAG, "... done reading from socket. Last read return=%d errno=%d.", r,
        close(s);
        for(int countdown = 10; countdown--) {
            ESP_LOGI(TAG, "%d... ", countdown);
            vTaskDelay(1000 / portTICK_PERIOD_MS);
        }
        ESP_LOGI(TAG, "Starting again!");
```

Also questioned it about 400 and 415 status error codes until I found the issue.

Lab 7 Part 3

Just combining previous code