

Embedded Systems

CS 397

TRIMESTER 3, AY 2021/22

# Hands-On 4-2

## Ethernet – LwIP TCP Echo Server

Dr. LIAW Hwee Choo

Department of Electrical and Computer Engineering

DigiPen Institute of Technology Singapore

HweeChoo.Liaw@DigiPen.edu

# Hands-On LwIP TCP Echo Server

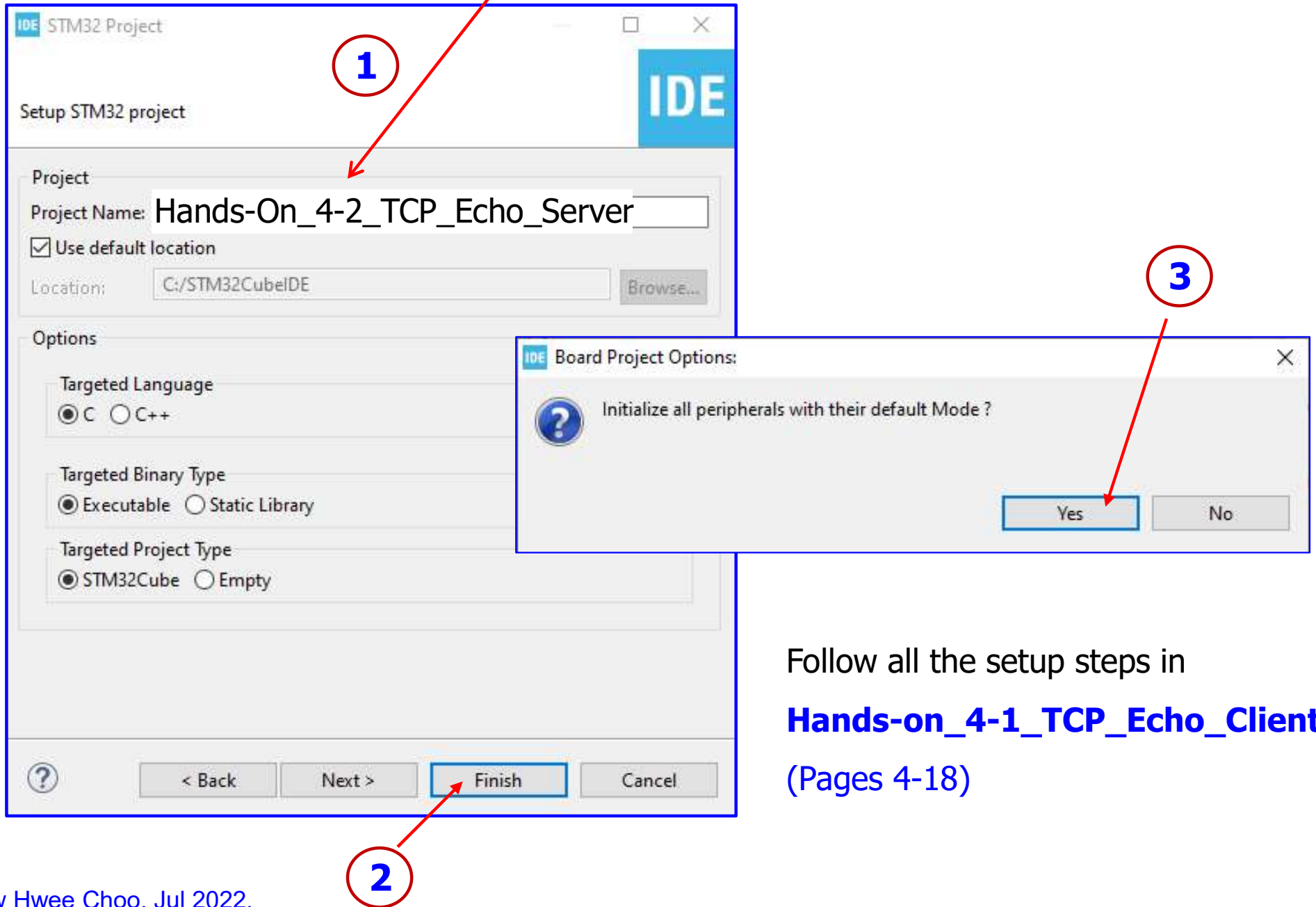
## Objectives

The aims of this hands-on session are to

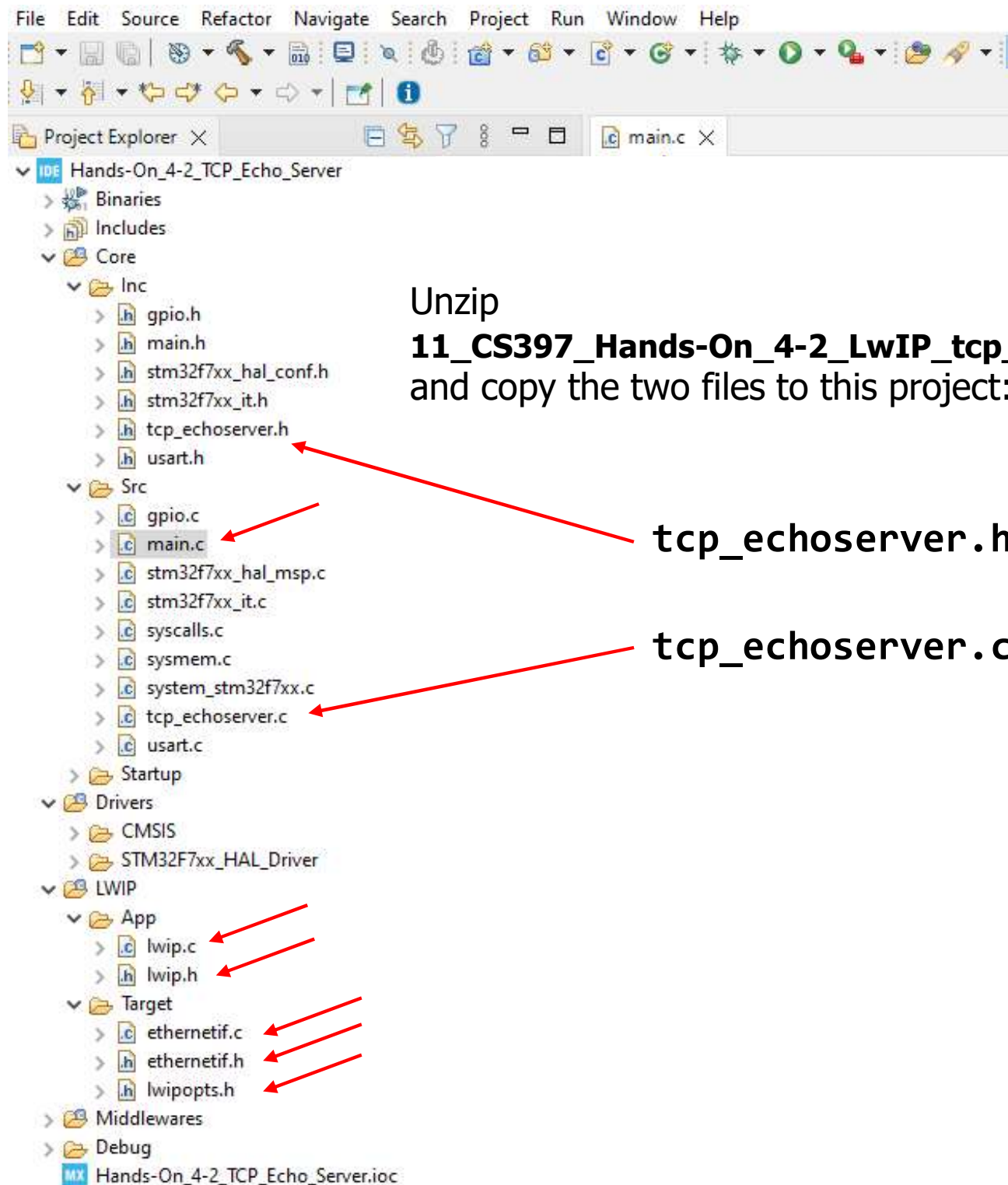
- implement a STM32 (STM32CubeIDE) project
- set up the Ethernet – LwIP TCP Echo Server application using STM32F767 microcontroller
- configure and program the Ethernet peripheral to make the microcontroller as a TCP echo server that waits for client requests
- test the developed program using the echotool.exe software
- build up the knowledge of Ethernet application development
  - Run [STM32CubeIDE](#)
  - [Select workspace: C:\STM32\\_CS397](#)
  - [File -> Close All Editors](#)
  - Start a [New STM32 Project](#)
  - Select the [Nucleo-F767ZI Board](#)

# Hands-On LwIP TCP Echo Server

Enter Project Name: [Hands-On\\_4-2\\_TCP\\_Echo\\_Server](#)



Follow all the setup steps in  
[Hands-on\\_4-1\\_TCP\\_Echo\\_Client](#)  
(Pages 4-18)



# Hands-On LwIP TCP Echo Server

## Add Code to **main.c**

```
/* Private includes */
/* USER CODE BEGIN Includes */

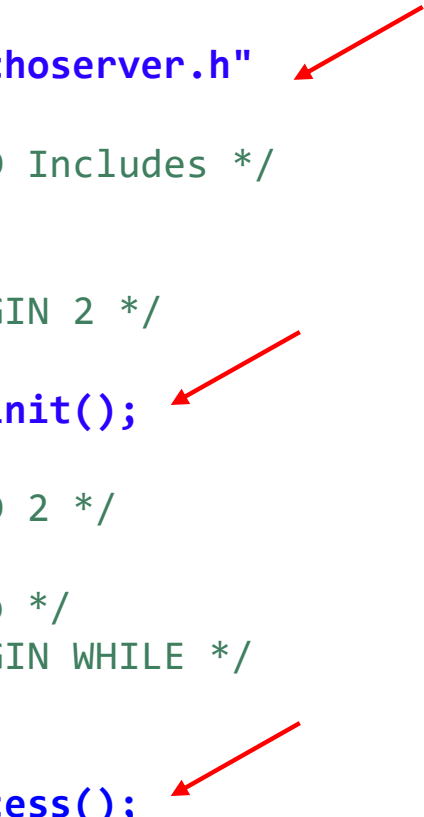
#include "tcp_echoserver.h"

/* USER CODE END Includes */

/* USER CODE BEGIN 2 */
tcp_echoserver_init();
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    MX_LWIP_Process();
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}
```



Purpose and Test procedure:

### **UM1713 User manual**

Developing applications on STM32Cube  
with LwIP TCP/IP stack

### **Section 6** Using the LwIP applications

#### 6.1.2 TCP Echo Server

## Hands-On LwIP TCP Echo Server

Modify Code in `tcp_echoserver.c`

```
/* Includes */  
#include "tcp_echoserver.h"  
#include "lwip/debug.h"  
#include "lwip/stats.h"  
#include "lwip/tcp.h"  
#include "main.h"
```

```
#define SERVER_PORT 7
```

Ensure this line is defined



This line is added to the original `tcp_echoserver.c`



# Hands-On LwIP TCP Echo Server

## Generated Code in Lwip.c

```
/* LwIP initialization function */
```

```
void MX_LWIP_Init(void)
```

```
{
```

```
    /* IP addresses initialization */
```

```
    IP_ADDRESS[0] = 192;
```

```
    IP_ADDRESS[1] = 168;
```

```
    IP_ADDRESS[2] = 1;
```

```
    IP_ADDRESS[3] = 205;
```

```
    NETMASK_ADDRESS[0] = 255;
```

```
    NETMASK_ADDRESS[1] = 255;
```

```
    NETMASK_ADDRESS[2] = 255;
```

```
    NETMASK_ADDRESS[3] = 0;
```

```
    GATEWAY_ADDRESS[0] = 192;
```

```
    GATEWAY_ADDRESS[1] = 168;
```

```
    GATEWAY_ADDRESS[2] = 1;
```

```
    GATEWAY_ADDRESS[3] = 1;
```

```
/* USER CODE BEGIN IP_ADDRESSES */
```

```
/* USER CODE END IP_ADDRESSES */
```

```
/* Initialize the LwIP stack without RTOS */
```

```
lwip_init();
```

```
/* IP addresses initialization without DHCP (IPv4) */
```

```
IP4_ADDR(&ipaddr, IP_ADDRESS[0], IP_ADDRESS[1], IP_ADDRESS[2], IP_ADDRESS[3]);
```

```
IP4_ADDR(&netmask, NETMASK_ADDRESS[0], NETMASK_ADDRESS[1], NETMASK_ADDRESS[2], NETMASK_ADDRESS[3]);
```

```
IP4_ADDR(&gw, GATEWAY_ADDRESS[0], GATEWAY_ADDRESS[1], GATEWAY_ADDRESS[2], GATEWAY_ADDRESS[3]);
```

```
/* add the network interface (IPv4/IPv6) without RTOS */
```

```
netif_add(&gnetif, &ipaddr, &netmask, &gw, NULL, &ethernetif_init, &ethernet_input);
```

For a different router (gateway):

```
IP_ADDRESS[0] = 192;
```

```
IP_ADDRESS[1] = 168;
```

```
IP_ADDRESS[2] = 50;
```

```
IP_ADDRESS[3] = 205;
```

```
NETMASK_ADDRESS[0] = 255;
```

```
NETMASK_ADDRESS[1] = 255;
```

```
NETMASK_ADDRESS[2] = 255;
```

```
NETMASK_ADDRESS[3] = 0;
```

```
GATEWAY_ADDRESS[0] = 192;
```

```
GATEWAY_ADDRESS[1] = 168;
```

```
GATEWAY_ADDRESS[2] = 50;
```

```
GATEWAY_ADDRESS[3] = 1;
```

# Hands-On LwIP TCP Echo Server

## Application Testing – LwIP TCP Echo Server

This application is used to test a basic TCP connection. The STM32 MCU acts as a TCP server that waits for clients request. It simply echoes back whatever is sent.

To test the TCP echo server application, follow these steps:

1. Build and program the project code into the STM32F767ZI Flash memory.
2. On the remote PC, open a command prompt window and go to **c:\CS397**
3. At the command prompt, enter:

**C:\CS397>echotool IP\_address /p tcp /r 7 /n 15 /t 2 /d Testing LwIP TCP echo server**

where:

- IP\_address is the [MCU board IP address](#).
- /p tcp is the protocol (TCP protocol)
- /r is the actual remote port on the echo server (echo port)
- /n is the number of echo requests (for example, 15)
- /t is the connection timeout in seconds (for example, 2)
- /d is the message to be sent for echo (for example, "Testing LwIP TCP echo server")



# Hands-On LwIP TCP Echo Server

Note:

1. Create another application with the dynamic address allocation "DHCP" for testing.
2. For "DHCP" application, check the router or "Advanced IP Scanner" for the MCU board IP address.
3. Figure shows an example of the command string and the module response.

The Results of LwIP  
TCP Echo Server

```
Command Prompt
6:5c8d%14
IPv4 Address. . . . . : 192.168.1.175
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

C:\CS397>ping 192.168.1.205

Pinging 192.168.1.205 with 32 bytes of data:
Reply from 192.168.1.205: bytes=32 time=2ms TTL=255
Reply from 192.168.1.205: bytes=32 time=1ms TTL=255
Reply from 192.168.1.205: bytes=32 time=1ms TTL=255
Reply from 192.168.1.205: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.1.205:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\CS397>echotool 192.168.1.205 /p tcp /r 7 /n 15 /t 2 /d
Testing LwIP TCP echo server

Hostname 192.168.1.205 resolved as 192.168.1.205

Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK
Reply from 192.168.1.205:7, time 1 ms OK
Reply from 192.168.1.205:7, time 0 ms OK

Statistics: Received=15, Corrupted=0
```