Embedded Systems

CS 397

TRIMESTER 3, AY 2021/22

Hands-On 4-4
Ethernet – LwIP UDP Echo Server

Dr. LIAW Hwee Choo

Department of Electrical and Computer Engineering

DigiPen Institute of Technology Singapore

HweeChoo.Liaw@DigiPen.edu

Hands-On LwIP UDP Echo Server

Objectives

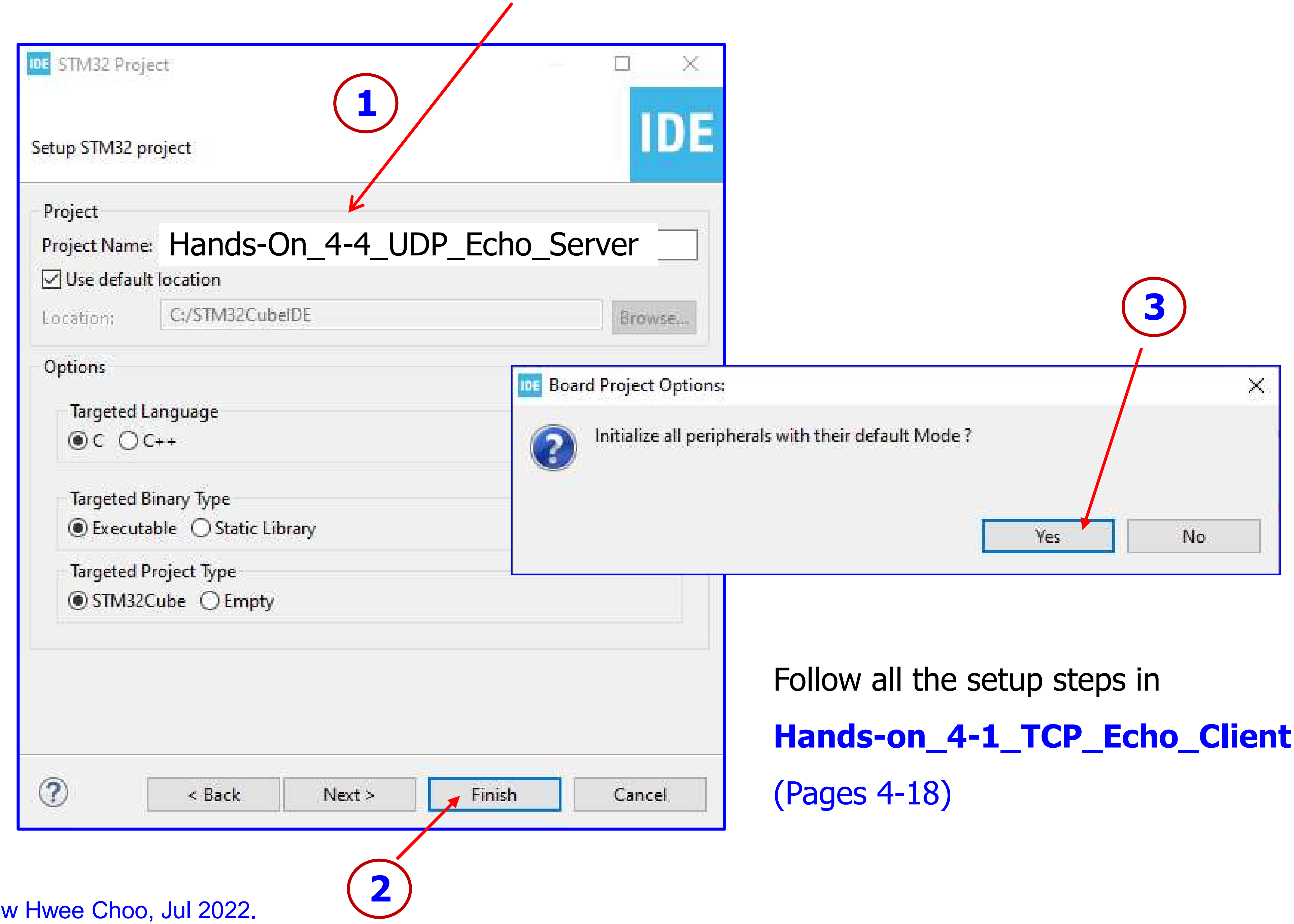
The aims of this hands-on session are to

- implement a STM32 (STM32CubeIDE) project
- set up the Ethernet – LwIP UDP Echo Server application using STM32F767 microcontroller
- configure and program the Ethernet peripheral to make the microcontroller as a UDP 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 UDP Echo Server

Enter Project Name: Hands-On_4-4_UDP_Echo_Server



STM32 Project

Setup STM32 project

Project

Project Name: Hands-On_4-4_UDP_Echo_Server

☑ Use default location

Location: C:/STM32CubeIDE Browse...

Options

Targeted Language
● C ○ C++

Targeted Binary Type
● Executable ○ Static Library

Targeted Project Type
● STM32Cube ○ Empty

< Back | Next > | Finish | Cancel

Board Project Options:

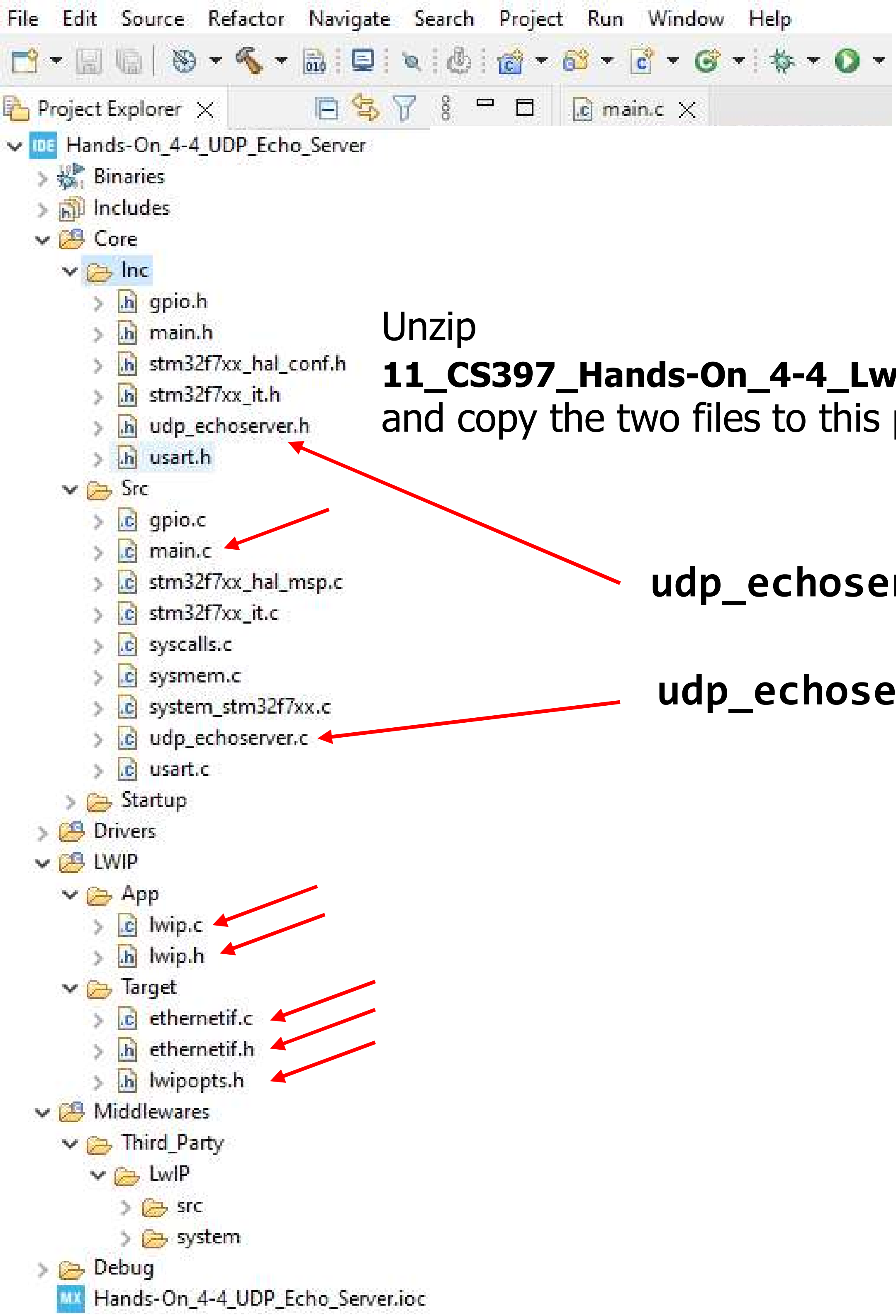
Initialize all peripherals with their default Mode ?

Yes | No

Follow all the setup steps in

Hands-on_4-1_TCP_Echo_Client

(Pages 4-18)



- Hands-On_4-4_UDP_Echo_Server
  - Binaries
  - Includes
  - Core
    - Inc
      - gpio.h
      - main.h
      - stm32f7xx_hal_conf.h
      - stm32f7xx_it.h
      - udp_echoserver.h
      - usart.h
    - Src
      - gpio.c
      - main.c
      - stm32f7xx_hal_msp.c
      - stm32f7xx_it.c
      - syscalls.c
      - sysmem.c
      - system_stm32f7xx.c
      - udp_echoserver.c
      - usart.c
    - Startup
  - Drivers
  - LWIP
    - App
      - lwip.c
      - lwip.h
    - Target
      - ethernetif.c
      - ethernetif.h
      - lwipopts.h
  - Middlewares
    - Third_Party
      - LwIP
        - src
        - system
  - Debug
  - Hands-On_4-4_UDP_Echo_Server.ioc

Unzip
11_CS397_Hands-On_4-4_LwIP_udp_echoserver.zip,
and copy the two files to this project:

`udp_echoserver.h`

`udp_echoserver.c`

Hands-On LwIP UDP Echo Server

Add Code to **main.c**

```
/* Private includes */
/* USER CODE BEGIN Includes */
#include "udp_echoserver.h"
/* USER CODE END Includes */

/* Initialize all configured
peripherals */
  MX_GPIO_Init();
  MX_USART3_UART_Init();
  MX_LWIP_Init();

/* USER CODE BEGIN 2 */
udp_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

6.1.4 UDP Echo Server

Hands-On LwIP UDP Echo Server

Modify Code in **`udp_echoserver.c`**

```c
/* Includes */
#include "main.h"
#include "lwip/pbuf.h"
#include "lwip/udp.h"
#include "lwip/tcp.h"
#include <string.h>
#include <stdio.h>
#include "udp_echoserver.h"

/* Private typedef */
/* Private define  */
#define UDP_SERVER_PORT    7   /* define the UDP local connection port */
#define UDP_CLIENT_PORT    7   /* define the UDP remote connection port */
```

Ensure these lines are defined

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 */

  /* Initilialize 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;
```

Testing of the Application – LwIP UDP Echo Server

This application is used to test a basic UDP connection. The STM32 MCU acts as a UDP server that waits for clients request.

To test the UDP 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 udp /r 7 /l 7 /n 15 /t 2 /d Testing LwIP UDP echo server
```

where: – IP_address is the MCU board IP address.

– /p udp is the protocol (UDP protocol)

– /r is the actual remote port on the echo server (echo port)

– /l is the actual local port for the client

– /n is the number of echo requests (for example, 15)

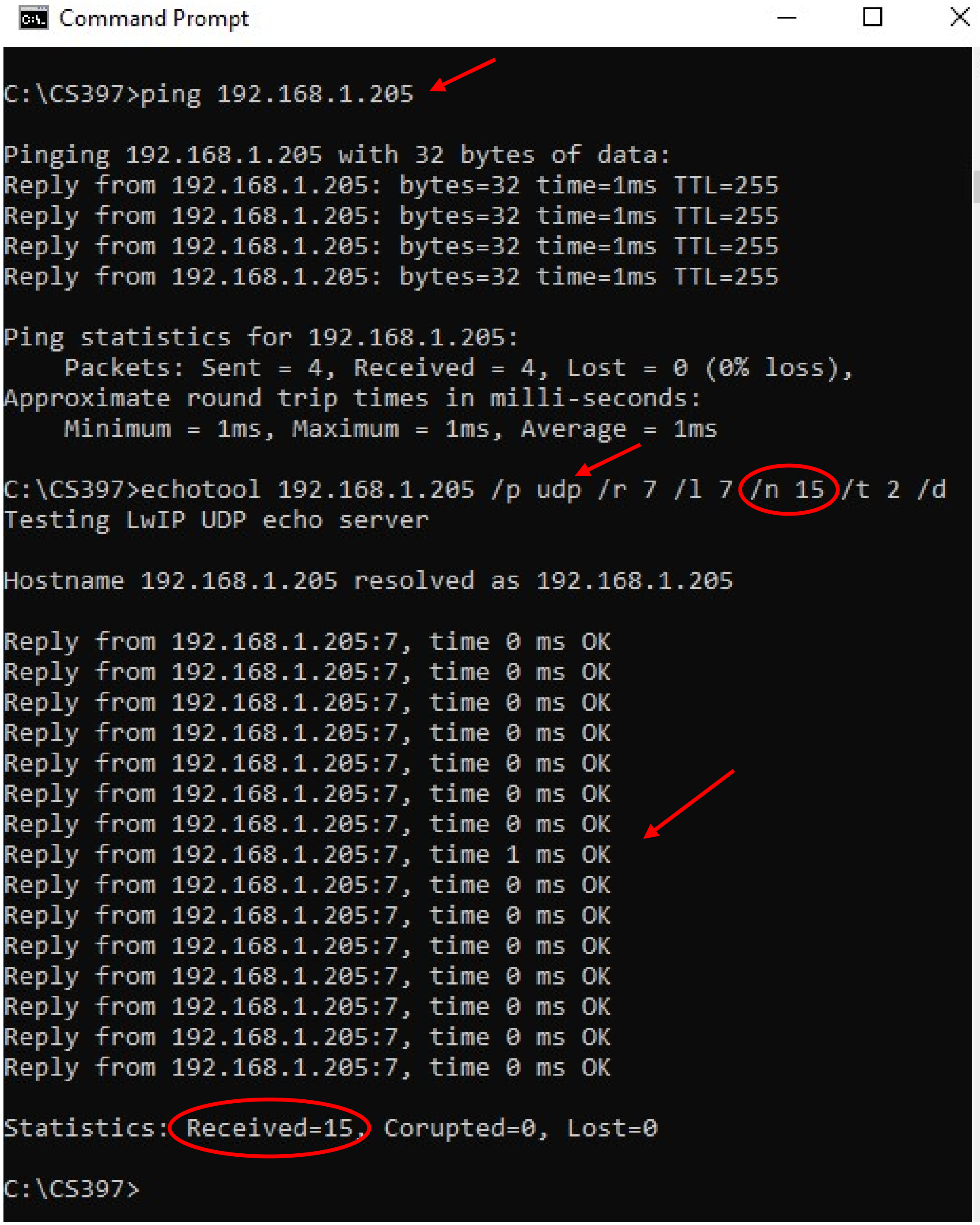
– /t is the connection timeout in seconds (for example, 2)

– /d is the message to be sent for echo (for example, “Testing LwIP UDP echo server”)

Hands-On LwIP UDP 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 this command string and the module response.



```
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=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
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 = 1ms, Average = 1ms

C:\CS397>echotool 192.168.1.205 /p udp /r 7 /l 7 /n 15 /t 2 /d
Testing LwIP UDP 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 1 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

Statistics: Received=15, Corupted=0, Lost=0

C:\CS397>
```

The Results of LwIP UDP Echo Server

- End -