Embedded Systems

CS 397

TRIMESTER 3, AY 2021/22

Hands-On 5-1
Ethernet – LwIP UDP TCP Echo Server Netconn RTOS

Dr. LIAW Hwee Choo

Department of Electrical and Computer Engineering

DigiPen Institute of Technology Singapore

HweeChoo.Liaw@DigiPen.edu

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

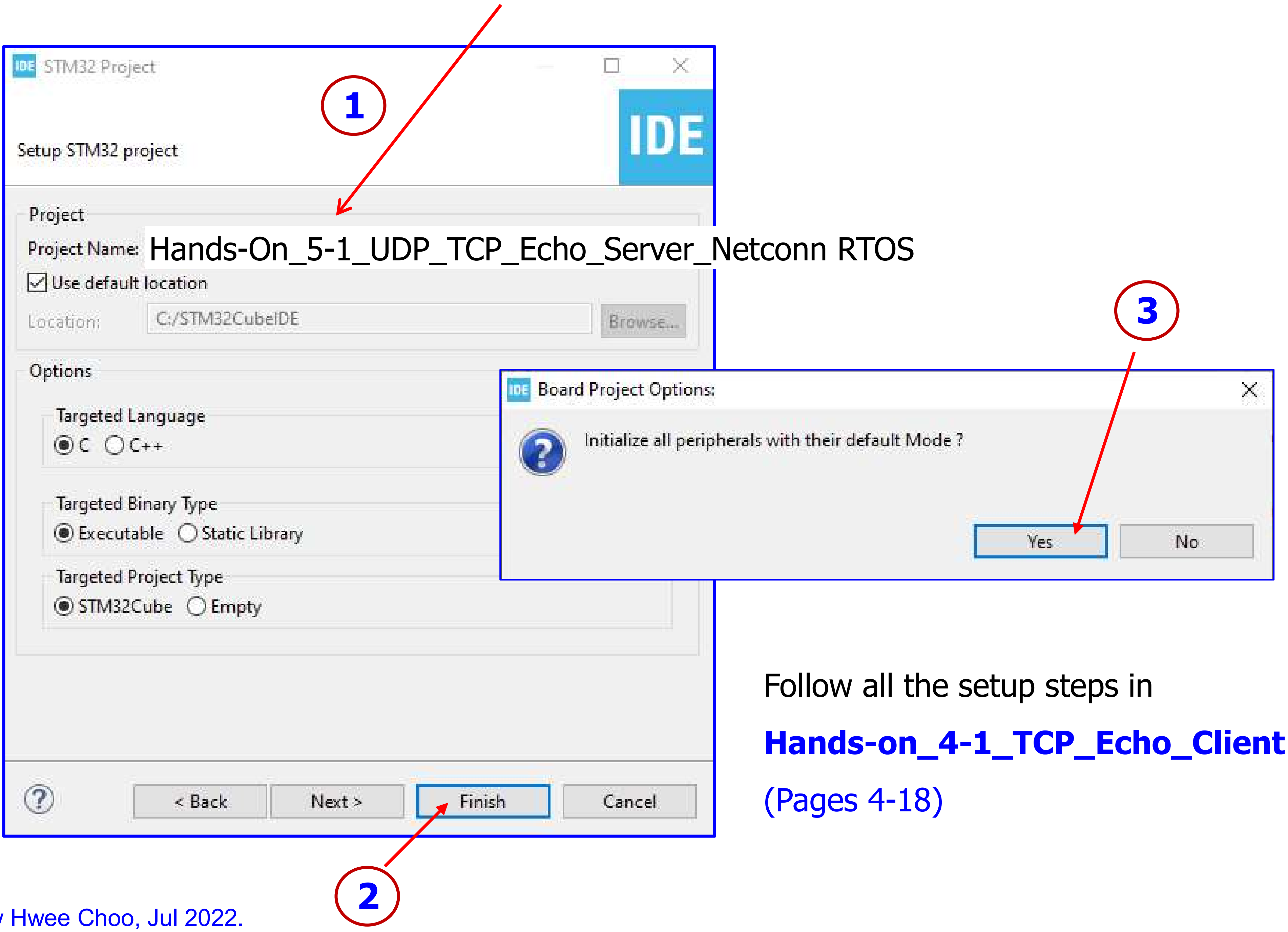
Objectives

The aims of this hands-on session are to

- implement a STM32 (STM32CubeIDE) project
- set up the Ethernet – LwIP UDP TCP echo server application based on Netconn RTOS using STM32F767 microcontroller
- configure and program the Ethernet peripheral to make the microcontroller as UDP and TCP echo servers 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 TCP Echo Server Netconn RTOS

Enter Project Name: Hands-On_5-1_UDP_TCP_Echo_Server_Netconn RTOS



Project Name: Hands-On_5-1_UDP_TCP_Echo_Server_Netconn RTOS

[✓] Use default location

Targeted Language: [✓ C] C++

Targeted Binary Type: [✓ Executable] Static Library

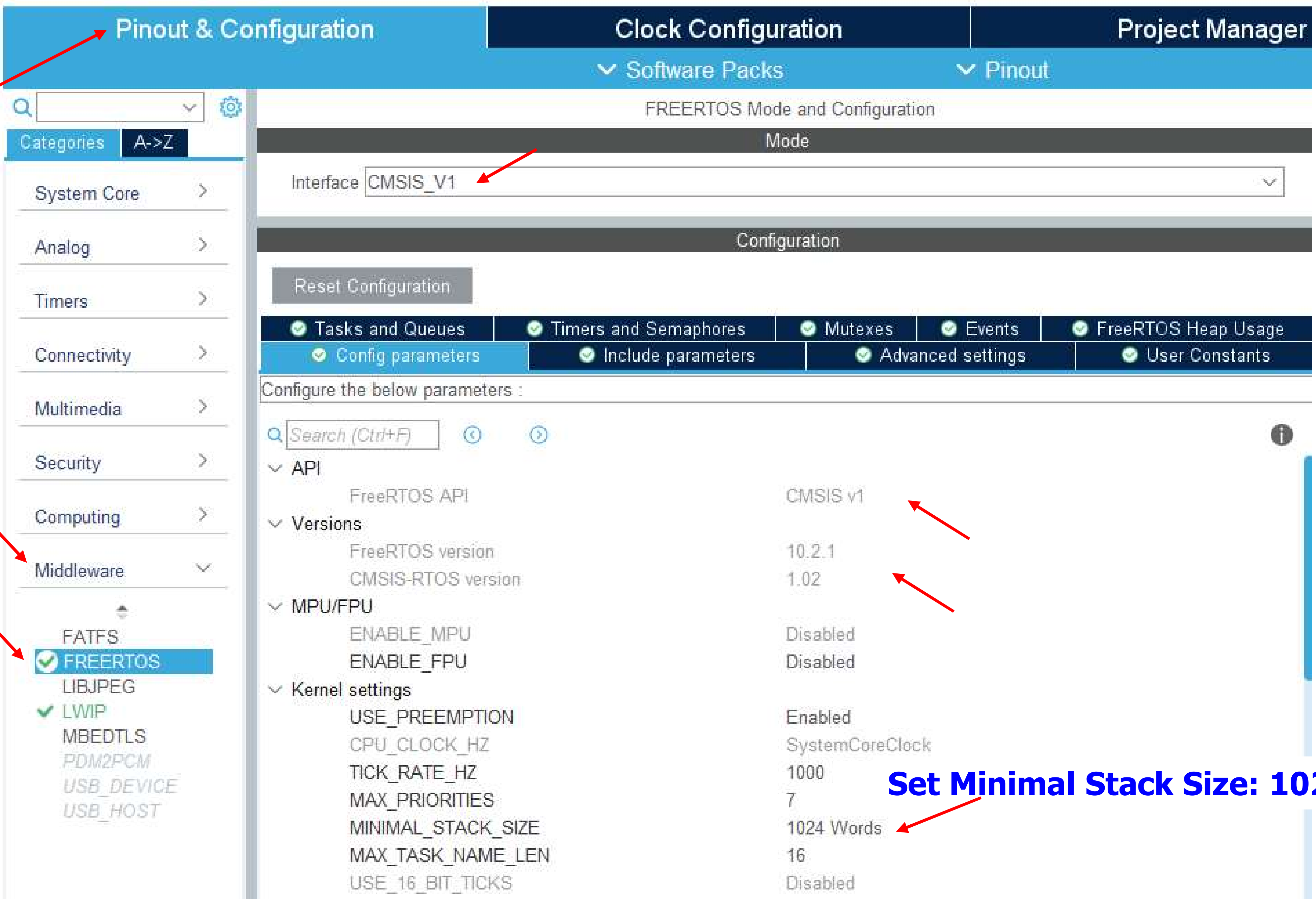
Targeted Project Type: [✓ STM32Cube] Empty

Follow all the setup steps in

Hands-on_4-1_TCP_Echo_Client

(Pages 4-18)

Enable **FREERTOS** by selecting the interface "**CMSIS_V1**".



FREERTOS Mode and Configuration

Interface CMSIS_V1

Configure the below parameters :

API
FreeRTOS API CMSIS v1

Versions
FreeRTOS version 10.2.1
CMSIS-RTOS version 1.02

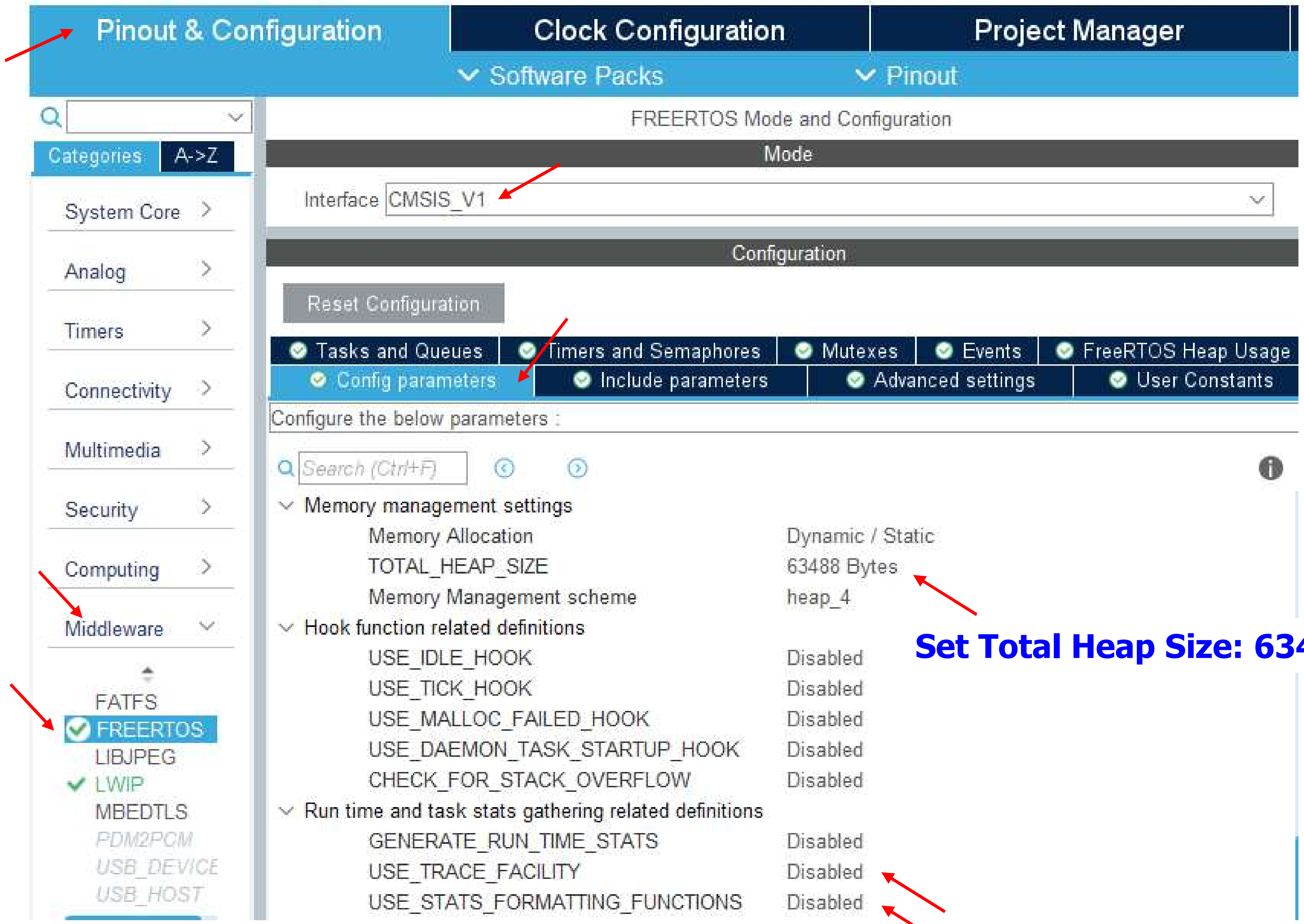
MPU/FPU
ENABLE_MPU Disabled
ENABLE_FPU Disabled

Kernel settings
USE_PREEMPTION Enabled
CPU_CLOCK_HZ SystemCoreClock
TICK_RATE_HZ 1000
MAX_PRIORITIES 7
MINIMAL_STACK_SIZE 1024 Words
MAX_TASK_NAME_LEN 16
USE_16_BIT_TICKS Disabled

Set Minimal Stack Size: 1024

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

Increase **TOTAL_HEAP_SIZE**



Interface CMSIS_V1

Configure the below parameters :

Memory management settings

Memory Allocation	Dynamic / Static
TOTAL_HEAP_SIZE	63488 Bytes
Memory Management scheme	heap_4

Hook function related definitions

USE_IDLE_HOOK	Disabled
USE_TICK_HOOK	Disabled
USE_MALLOC_FAILED_HOOK	Disabled
USE_DAEMON_TASK_STARTUP_HOOK	Disabled
CHECK_FOR_STACK_OVERFLOW	Disabled

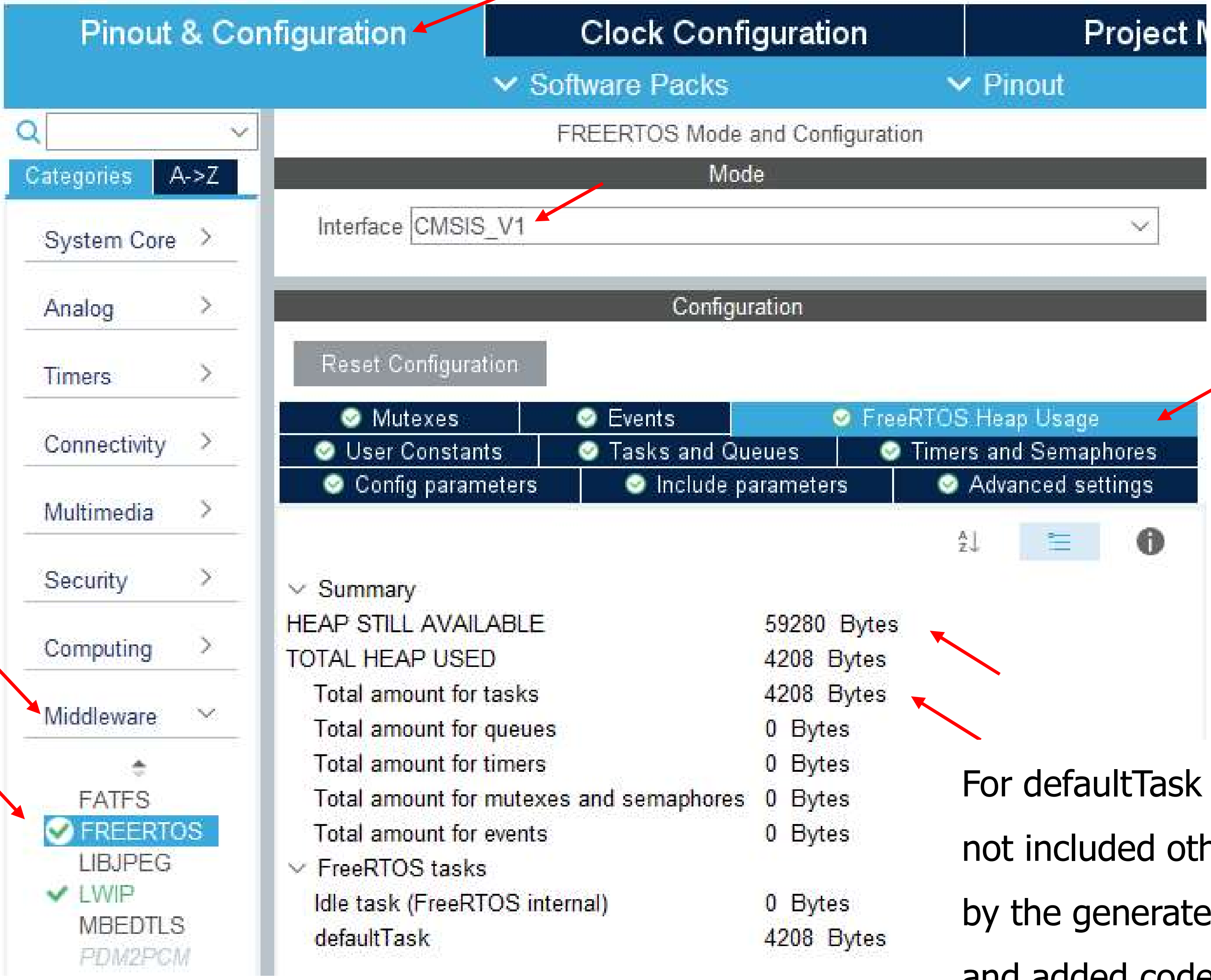
Run time and task stats gathering related definitions

GENERATE_RUN_TIME_STATS	Disabled
USE_TRACE_FACILITY	Disabled
USE_STATS_FORMATTING_FUNCTIONS	Disabled

Set Total Heap Size: 63488

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

FreeRTOS Heap Usage



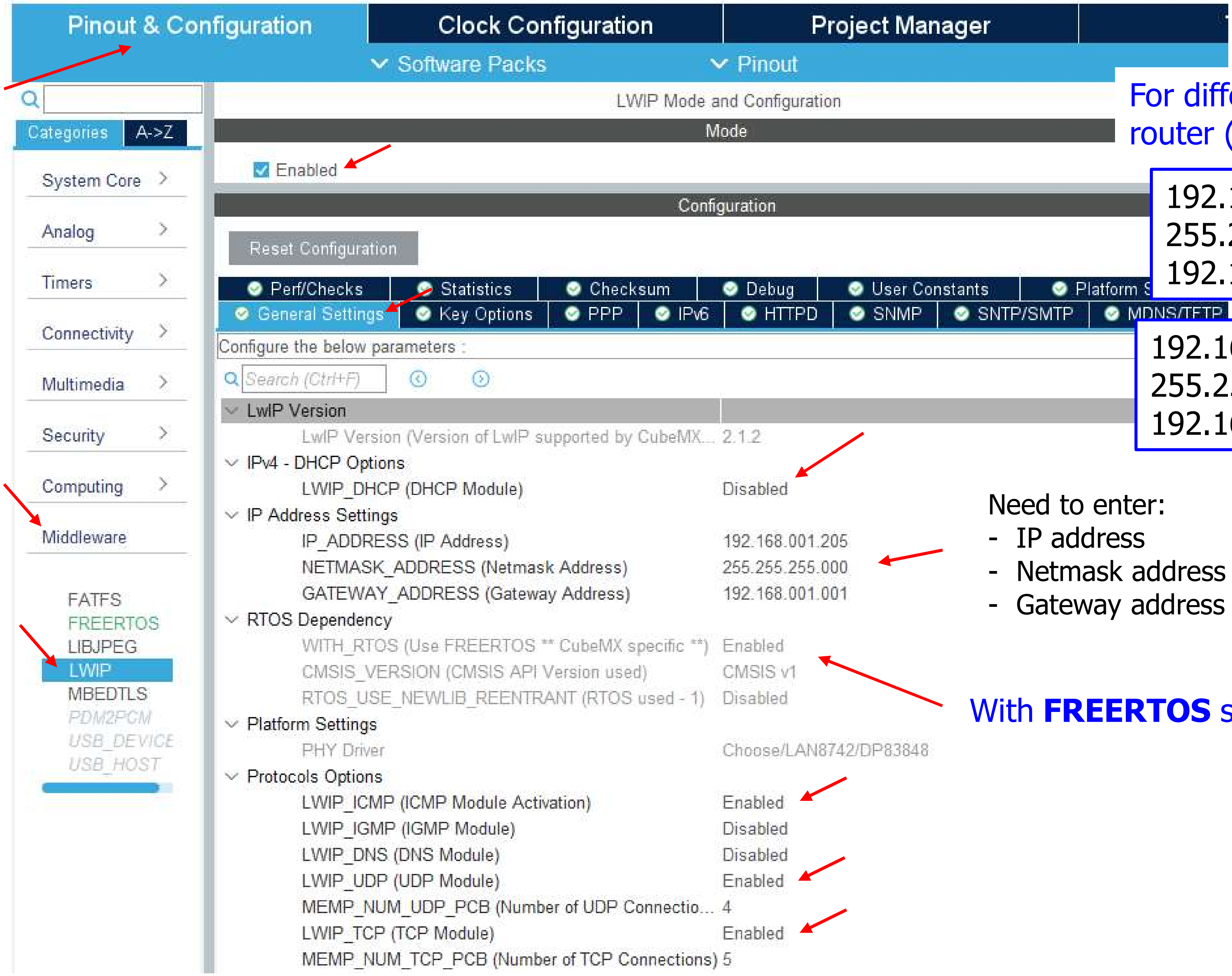
Interface CMSIS_V1

```
Summary
HEAP STILL AVAILABLE                            59280  Bytes
TOTAL HEAP USED                                 4208  Bytes
   Total amount for tasks                       4208  Bytes
   Total amount for queues                      0  Bytes
   Total amount for timers                      0  Bytes
   Total amount for mutexes and semaphores      0  Bytes
   Total amount for events                      0  Bytes
FreeRTOS tasks
   Idle task (FreeRTOS internal)                0  Bytes
   defaultTask                                  4208  Bytes
```

For defaultTask only, it is not included other tasks by the generated code and added code.

LwIP Settings

Hands-On LwIP UDP TCP Echo Server Netconn RTOS



LWIP Mode and Configuration

Enabled ✓

Configure the below parameters :

LwIP Version	
LwIP Version (Version of LwIP supported by CubeMX...	2.1.2
IPv4 - DHCP Options	
LWIP_DHCP (DHCP Module)	Disabled
IP Address Settings	
IP_ADDRESS (IP Address)	192.168.001.205
NETMASK_ADDRESS (Netmask Address)	255.255.255.000
GATEWAY_ADDRESS (Gateway Address)	192.168.001.001
RTOS Dependency	
WITH_RTOS (Use FREERTOS ** CubeMX specific **)	Enabled
CMSIS_VERSION (CMSIS API Version used)	CMSIS v1
RTOS_USE_NEWLIB_REENTRANT (RTOS used - 1)	Disabled
Platform Settings	
PHY Driver	Choose/LAN8742/DP83848
Protocols Options	
LWIP_ICMP (ICMP Module Activation)	Enabled
LWIP_IGMP (IGMP Module)	Disabled
LWIP_DNS (DNS Module)	Disabled
LWIP_UDP (UDP Module)	Enabled
MEMP_NUM_UDP_PCB (Number of UDP Connectio...	4
LWIP_TCP (TCP Module)	Enabled
MEMP_NUM_TCP_PCB (Number of TCP Connections)	5

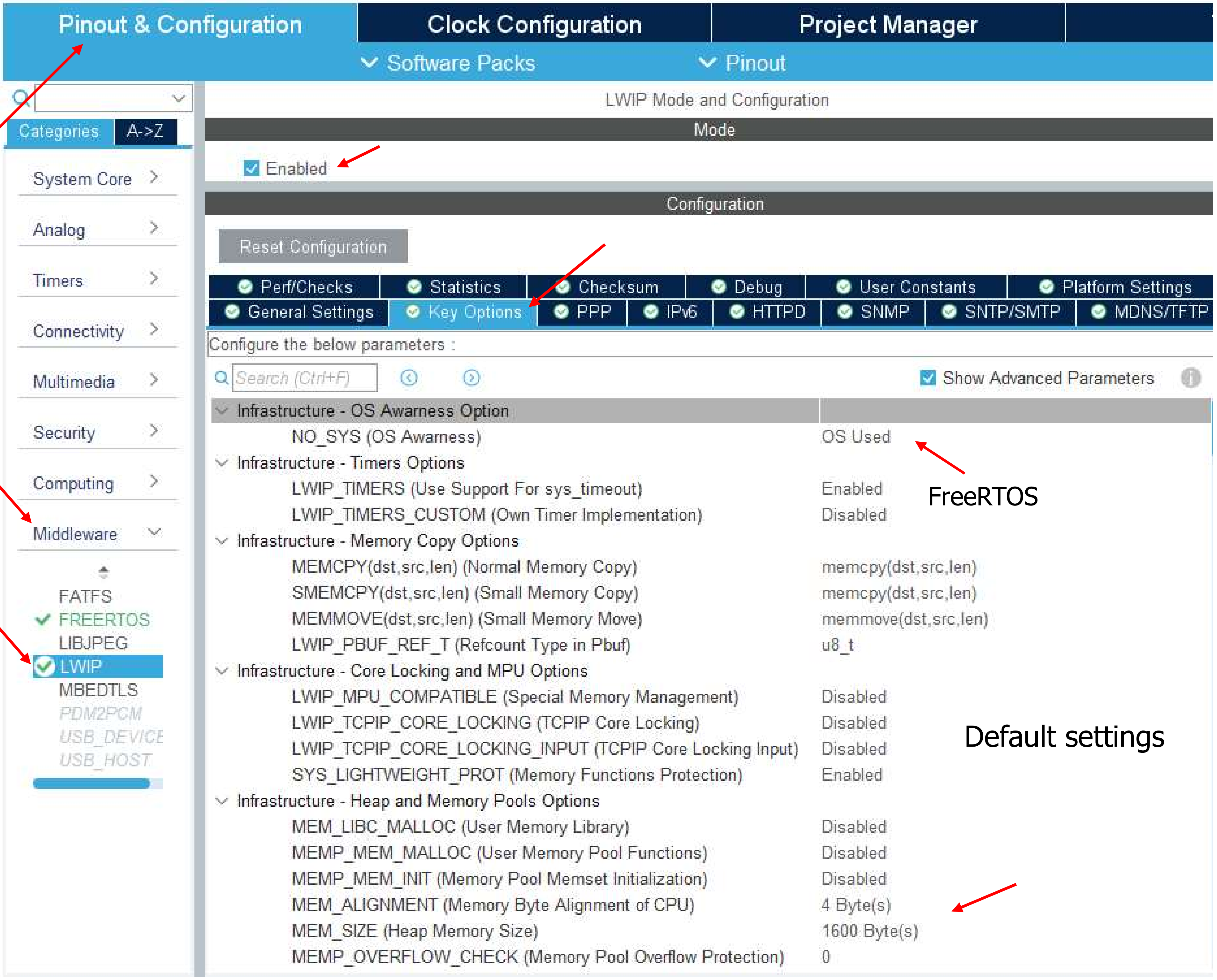
For different router (gateway):

192.168.1.205
255.255.255.0
192.168.1.1

192.168.50.205
255.255.255.0
192.168.50.1

Need to enter:
- IP address
- Netmask address
- Gateway address

With **FREERTOS** selected



LWIP Mode and Configuration

Mode

☑ Enabled

Configuration

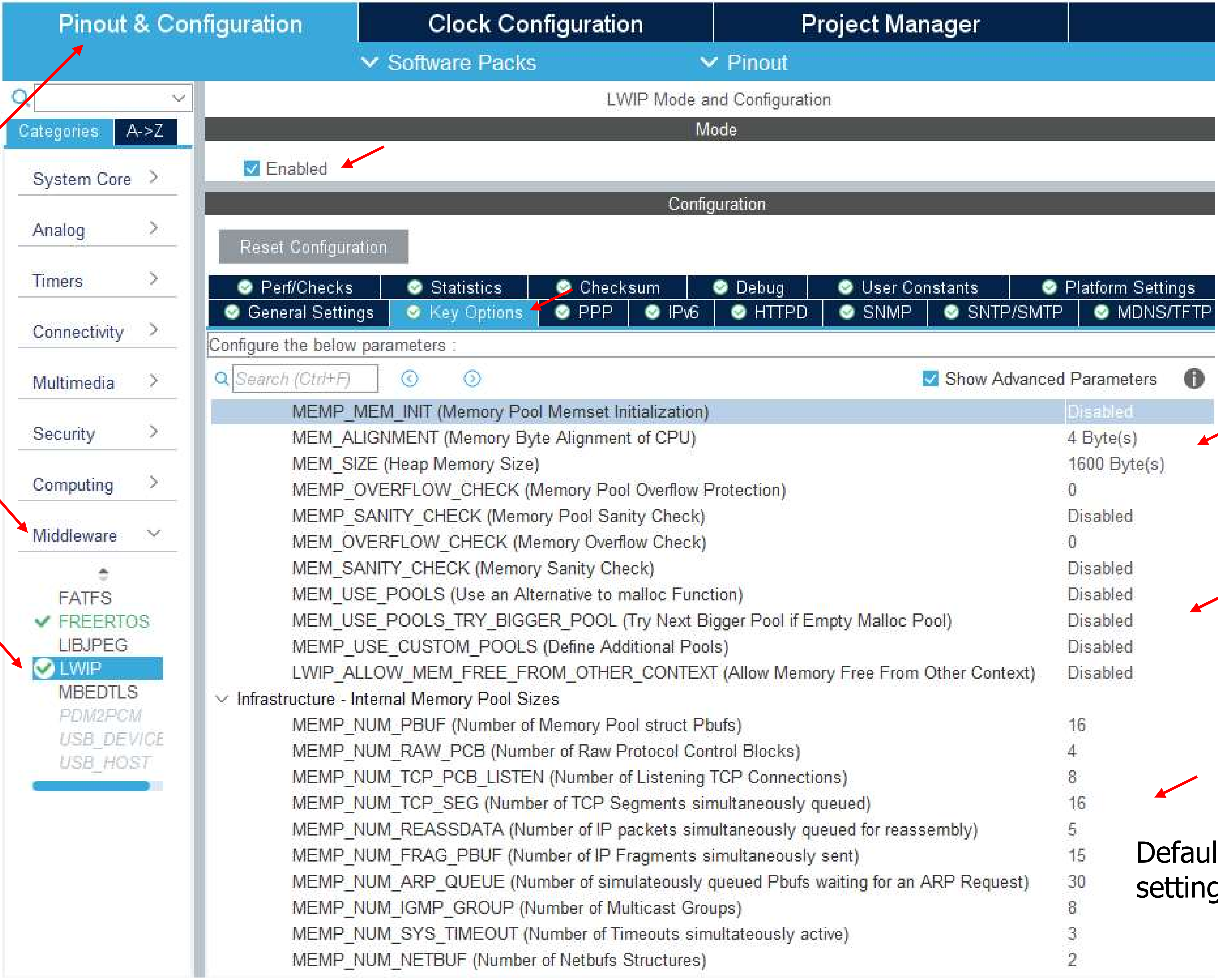
Configure the below parameters :

☑ Show Advanced Parameters

```
Infrastructure - OS Awarness Option
    NO_SYS (OS Awarness)                                      OS Used
Infrastructure - Timers Options
    LWIP_TIMERS (Use Support For sys_timeout)                 Enabled
    LWIP_TIMERS_CUSTOM (Own Timer Implementation)             Disabled
Infrastructure - Memory Copy Options
    MEMCPY(dst,src,len) (Normal Memory Copy)                  memcpy(dst,src,len)
    SMEMCPY(dst,src,len) (Small Memory Copy)                  memcpy(dst,src,len)
    MEMMOVE(dst,src,len) (Small Memory Move)                  memmove(dst,src,len)
    LWIP_PBUF_REF_T (Refcount Type in Pbuf)                   u8_t
Infrastructure - Core Locking and MPU Options
    LWIP_MPU_COMPATIBLE (Special Memory Management)           Disabled
    LWIP_TCPIP_CORE_LOCKING (TCPIP Core Locking)              Disabled
    LWIP_TCPIP_CORE_LOCKING_INPUT (TCPIP Core Locking Input)  Disabled
    SYS_LIGHTWEIGHT_PROT (Memory Functions Protection)        Enabled
Infrastructure - Heap and Memory Pools Options
    MEM_LIBC_MALLOC (User Memory Library)                     Disabled
    MEMP_MEM_MALLOC (User Memory Pool Functions)              Disabled
    MEMP_MEM_INIT (Memory Pool Memset Initialization)         Disabled
    MEM_ALIGNMENT (Memory Byte Alignment of CPU)              4 Byte(s)
    MEM_SIZE (Heap Memory Size)                               1600 Byte(s)
    MEMP_OVERFLOW_CHECK (Memory Pool Overflow Protection)     0
```

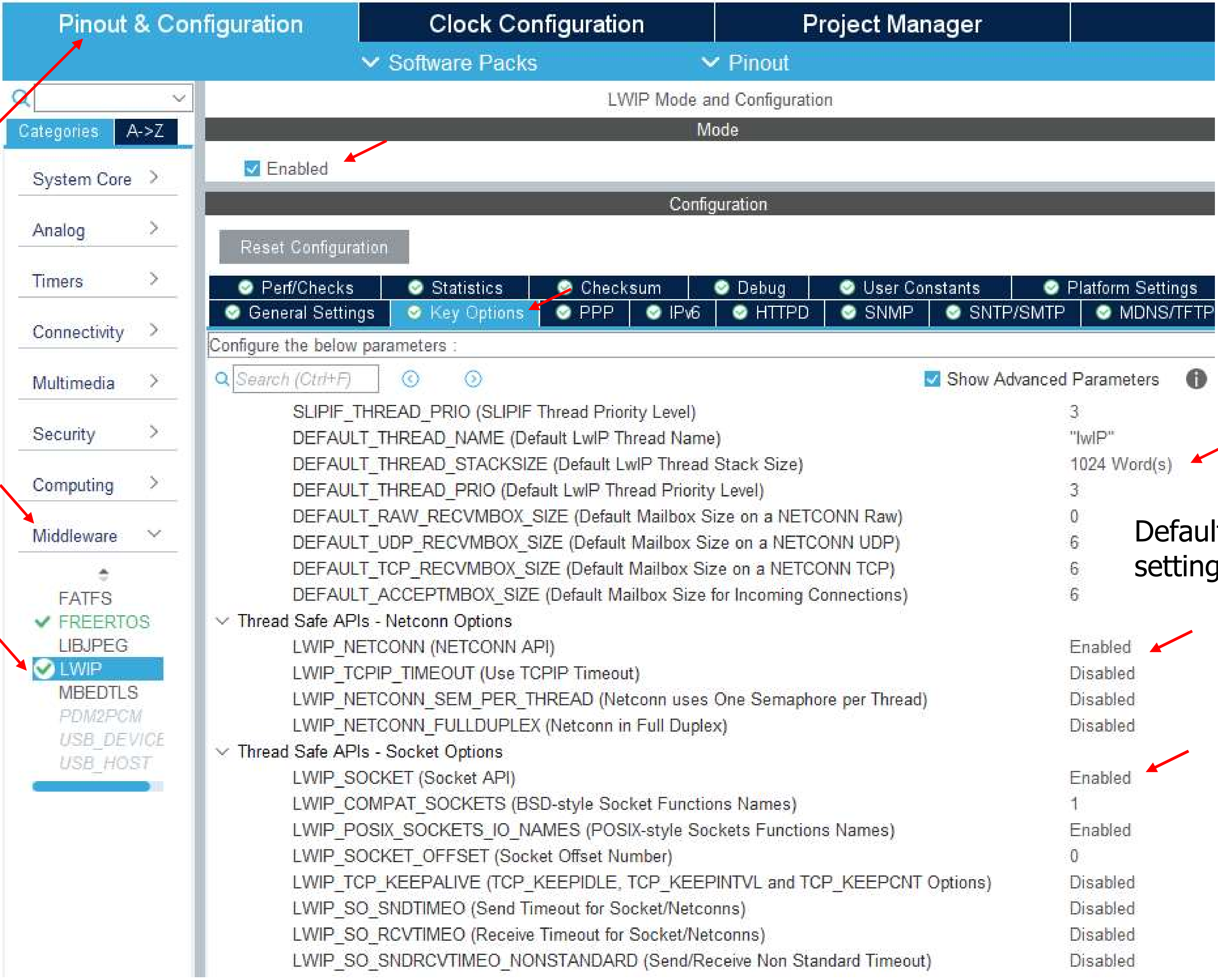
FreeRTOS

Default settings



MEMP_MEM_INIT (Memory Pool Memset Initialization)	Disabled
MEM_ALIGNMENT (Memory Byte Alignment of CPU)	4 Byte(s)
MEM_SIZE (Heap Memory Size)	1600 Byte(s)
MEMP_OVERFLOW_CHECK (Memory Pool Overflow Protection)	0
MEMP_SANITY_CHECK (Memory Pool Sanity Check)	Disabled
MEM_OVERFLOW_CHECK (Memory Overflow Check)	0
MEM_SANITY_CHECK (Memory Sanity Check)	Disabled
MEM_USE_POOLS (Use an Alternative to malloc Function)	Disabled
MEM_USE_POOLS_TRY_BIGGER_POOL (Try Next Bigger Pool if Empty Malloc Pool)	Disabled
MEMP_USE_CUSTOM_POOLS (Define Additional Pools)	Disabled
LWIP_ALLOW_MEM_FREE_FROM_OTHER_CONTEXT (Allow Memory Free From Other Context)	Disabled
Infrastructure - Internal Memory Pool Sizes	
MEMP_NUM_PBUF (Number of Memory Pool struct Pbufs)	16
MEMP_NUM_RAW_PCB (Number of Raw Protocol Control Blocks)	4
MEMP_NUM_TCP_PCB_LISTEN (Number of Listening TCP Connections)	8
MEMP_NUM_TCP_SEG (Number of TCP Segments simultaneously queued)	16
MEMP_NUM_REASSDATA (Number of IP packets simultaneously queued for reassembly)	5
MEMP_NUM_FRAG_PBUF (Number of IP Fragments simultaneously sent)	15
MEMP_NUM_ARP_QUEUE (Number of simulateously queued Pbufs waiting for an ARP Request)	30
MEMP_NUM_IGMP_GROUP (Number of Multicast Groups)	8
MEMP_NUM_SYS_TIMEOUT (Number of Timeouts simultateously active)	3
MEMP_NUM_NETBUF (Number of Netbufs Structures)	2

Default settings



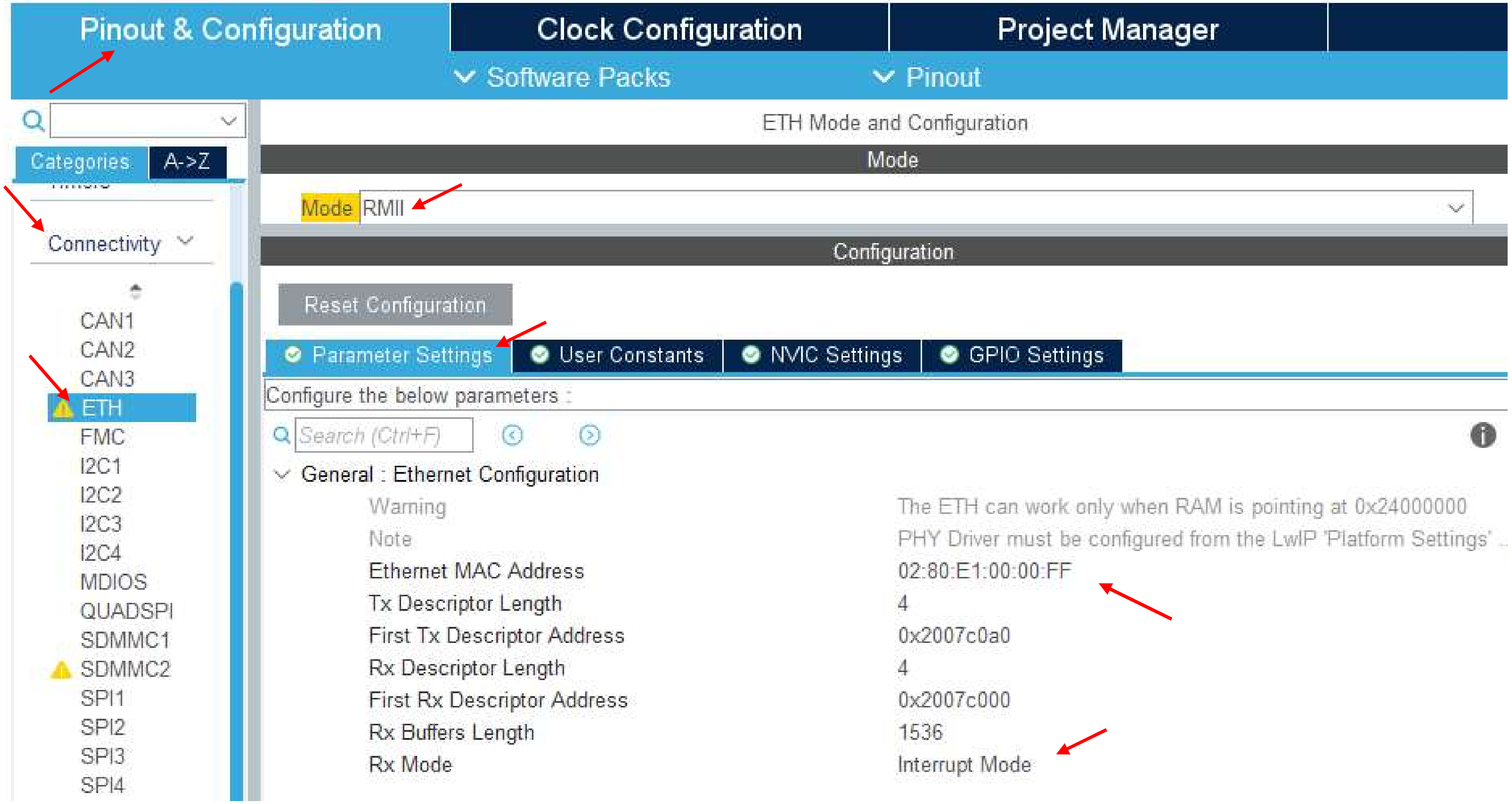
[✓] Enabled

[✓] Show Advanced Parameters

SLIPIF_THREAD_PRIO (SLIPIF Thread Priority Level)	3
DEFAULT_THREAD_NAME (Default LwIP Thread Name)	"lwIP"
DEFAULT_THREAD_STACKSIZE (Default LwIP Thread Stack Size)	1024 Word(s)
DEFAULT_THREAD_PRIO (Default LwIP Thread Priority Level)	3
DEFAULT_RAW_RECVMBOX_SIZE (Default Mailbox Size on a NETCONN Raw)	0
DEFAULT_UDP_RECVMBOX_SIZE (Default Mailbox Size on a NETCONN UDP)	6
DEFAULT_TCP_RECVMBOX_SIZE (Default Mailbox Size on a NETCONN TCP)	6
DEFAULT_ACCEPTMBOX_SIZE (Default Mailbox Size for Incoming Connections)	6
Thread Safe APIs - Netconn Options	
LWIP_NETCONN (NETCONN API)	Enabled
LWIP_TCPIP_TIMEOUT (Use TCPIP Timeout)	Disabled
LWIP_NETCONN_SEM_PER_THREAD (Netconn uses One Semaphore per Thread)	Disabled
LWIP_NETCONN_FULLDUPLEX (Netconn in Full Duplex)	Disabled
Thread Safe APIs - Socket Options	
LWIP_SOCKET (Socket API)	Enabled
LWIP_COMPAT_SOCKETS (BSD-style Socket Functions Names)	1
LWIP_POSIX_SOCKETS_IO_NAMES (POSIX-style Sockets Functions Names)	Enabled
LWIP_SOCKET_OFFSET (Socket Offset Number)	0
LWIP_TCP_KEEPALIVE (TCP_KEEPIDLE, TCP_KEEPINTVL and TCP_KEEPCNT Options)	Disabled
LWIP_SO_SNDTIMEO (Send Timeout for Socket/Netconns)	Disabled
LWIP_SO_RCVTIMEO (Receive Timeout for Socket/Netconns)	Disabled
LWIP_SO_SNDRCVTIMEO_NONSTANDARD (Send/Receive Non Standard Timeout)	Disabled

Default settings

With **FREERTOS** selected, **Ethernet Basic Configuration** is modified.



ETH Mode and Configuration

Mode

Mode RMII

Configuration

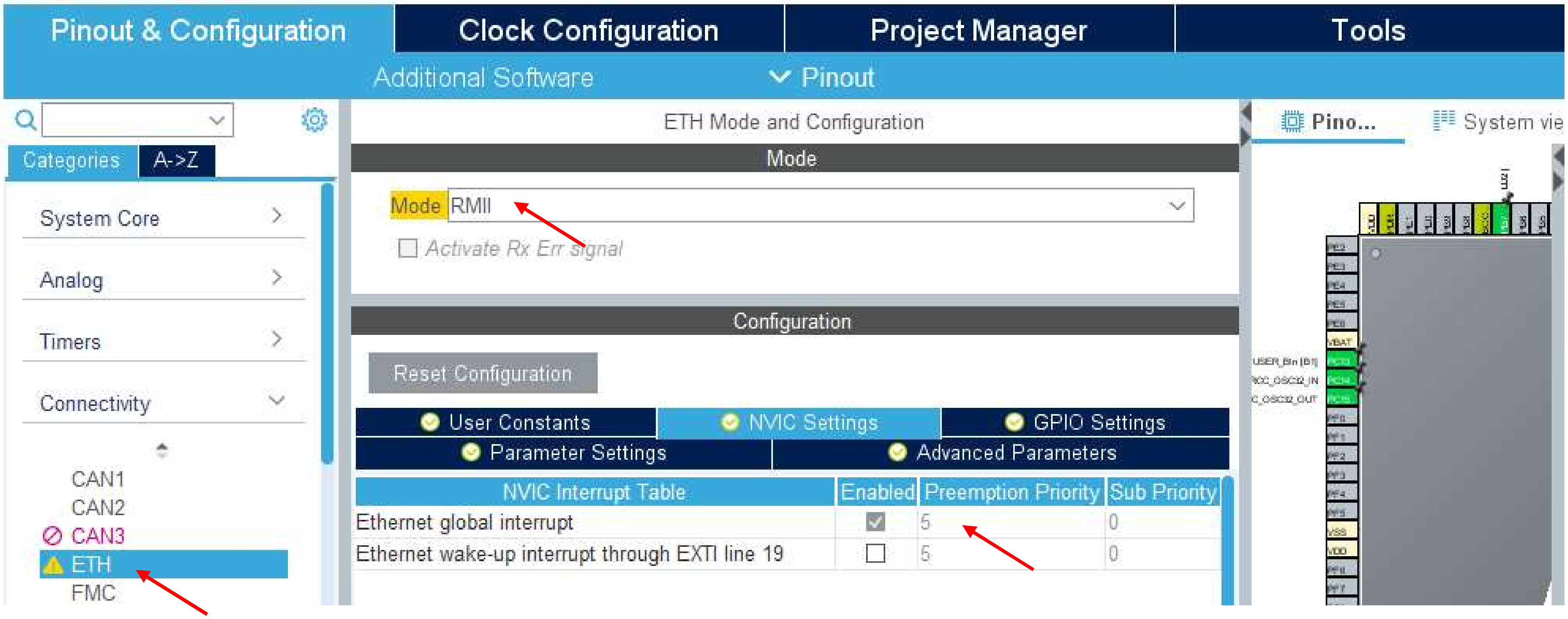
Configure the below parameters :

General : Ethernet Configuration

Warning	The ETH can work only when RAM is pointing at 0x24000000
Note	PHY Driver must be configured from the LwIP 'Platform Settings'
Ethernet MAC Address	02:80:E1:00:00:FF
Tx Descriptor Length	4
First Tx Descriptor Address	0x2007c0a0
Rx Descriptor Length	4
First Rx Descriptor Address	0x2007c000
Rx Buffers Length	1536
Rx Mode	Interrupt Mode

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

With FREERTOS selected, Ethernet Global Interrupt is enabled and assigned with Preemption Priority.

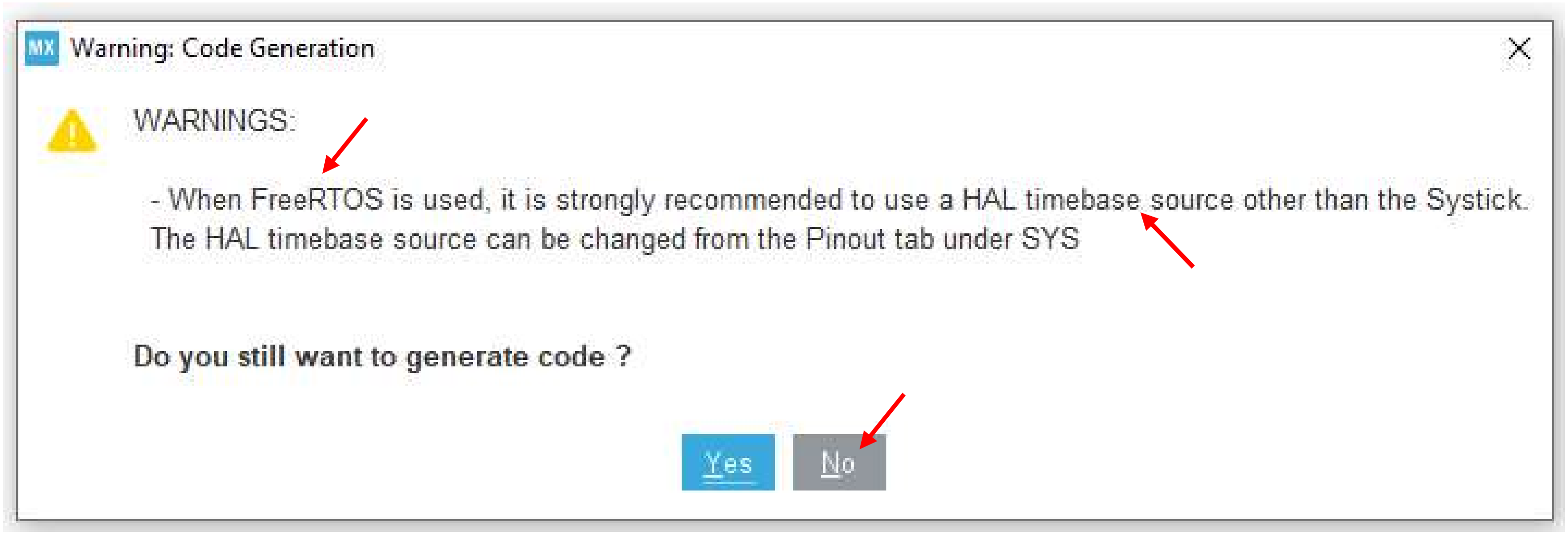


Mode: RMII

☐ Activate Rx Err signal

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Ethernet global interrupt	✓	5	0
Ethernet wake-up interrupt through EXTI line 19	☐	5	0

With **FREERTOS** selected, the **Timebase Source** is changed to **TIM7** manually.

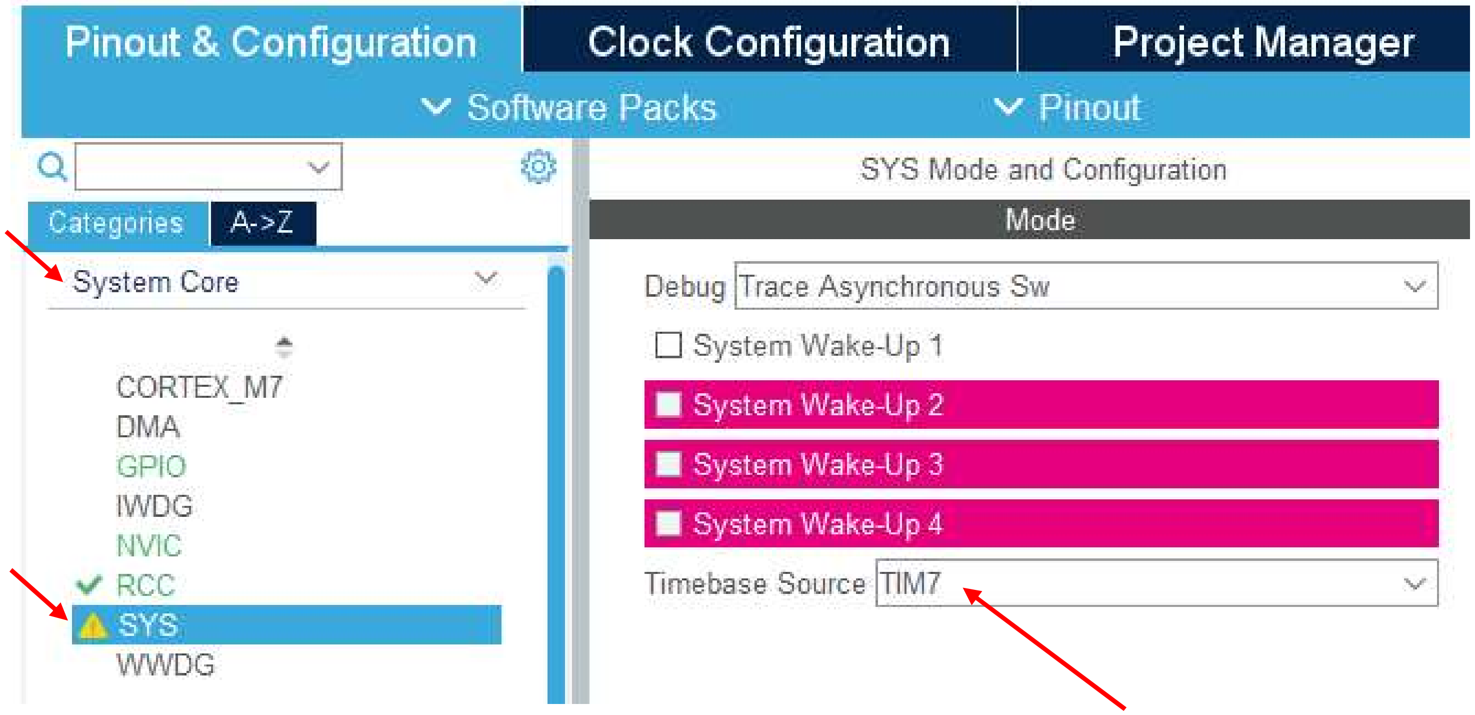


Warning: Code Generation

WARNINGS:

- When FreeRTOS is used, it is strongly recommended to use a HAL timebase source other than the Systick. The HAL timebase source can be changed from the Pinout tab under SYS

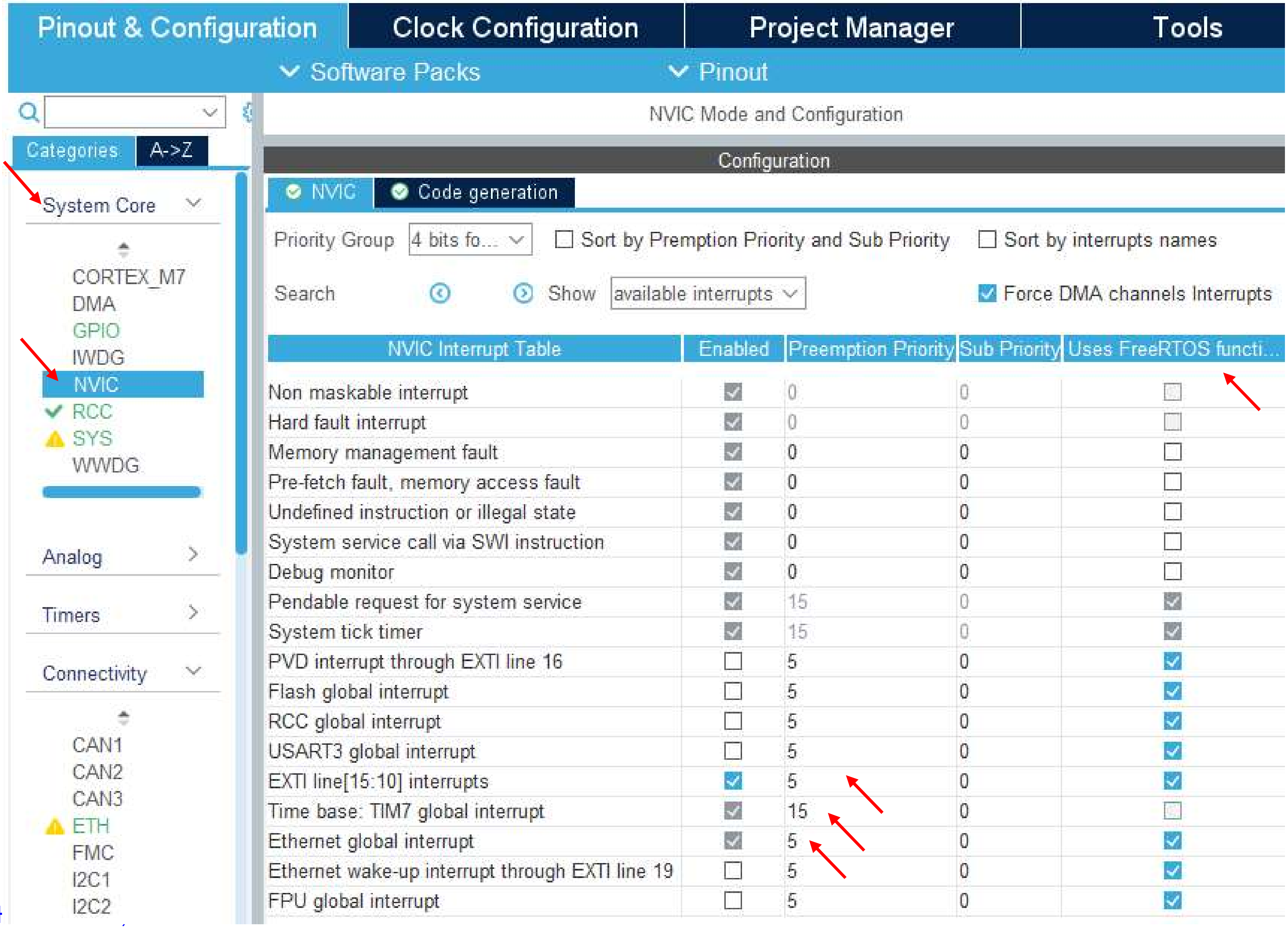
Do you still want to generate code ?



Debug: Trace Asynchronous Sw
System Wake-Up 1
System Wake-Up 2
System Wake-Up 3
System Wake-Up 4
Timebase Source: TIM7

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

With **FREERTOS** and **Time Base** selections, the NVIC settings are modified automatically



Priority Group: 4 bits fo...

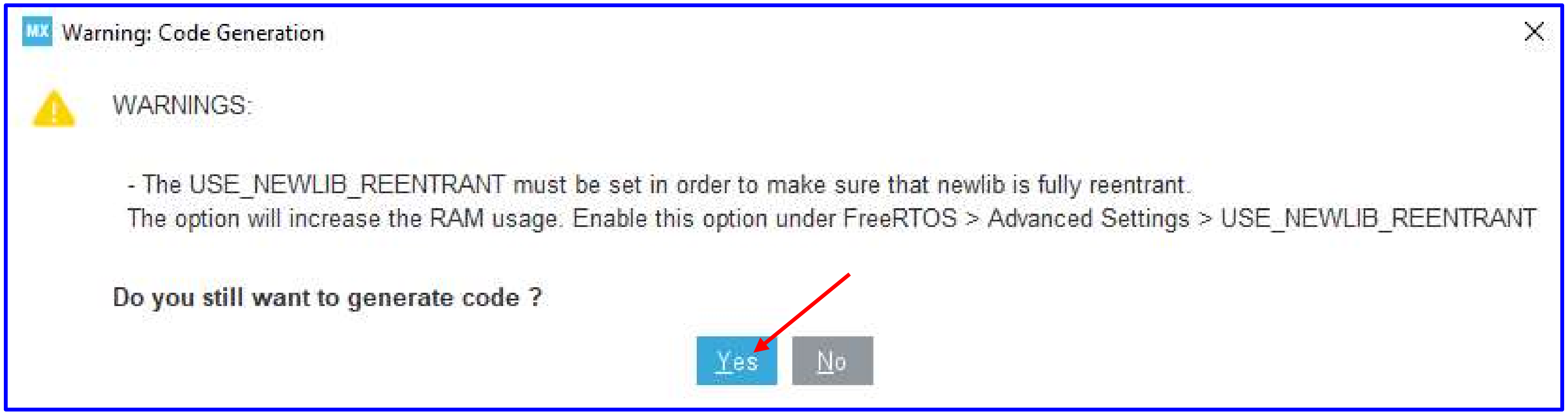
☐ Sort by Premption Priority and Sub Priority ☐ Sort by interrupts names

Search Show: available interrupts ☑ Force DMA channels Interrupts

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority	Uses FreeRTOS functi...
Non maskable interrupt	✓	0	0	☐
Hard fault interrupt	✓	0	0	☐
Memory management fault	✓	0	0	☐
Pre-fetch fault, memory access fault	✓	0	0	☐
Undefined instruction or illegal state	✓	0	0	☐
System service call via SWI instruction	✓	0	0	☐
Debug monitor	✓	0	0	☐
Pendable request for system service	✓	15	0	✓
System tick timer	✓	15	0	✓
PVD interrupt through EXTI line 16	☐	5	0	✓
Flash global interrupt	☐	5	0	✓
RCC global interrupt	☐	5	0	✓
USART3 global interrupt	☐	5	0	✓
EXTI line[15:10] interrupts	✓	5	0	✓
Time base: TIM7 global interrupt	✓	15	0	☐
Ethernet global interrupt	✓	5	0	✓
Ethernet wake-up interrupt through EXTI line 19	☐	5	0	✓
FPU global interrupt	☐	5	0	✓

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

Code Generation: Do not enable USE_NEWLIB_REENTRANT

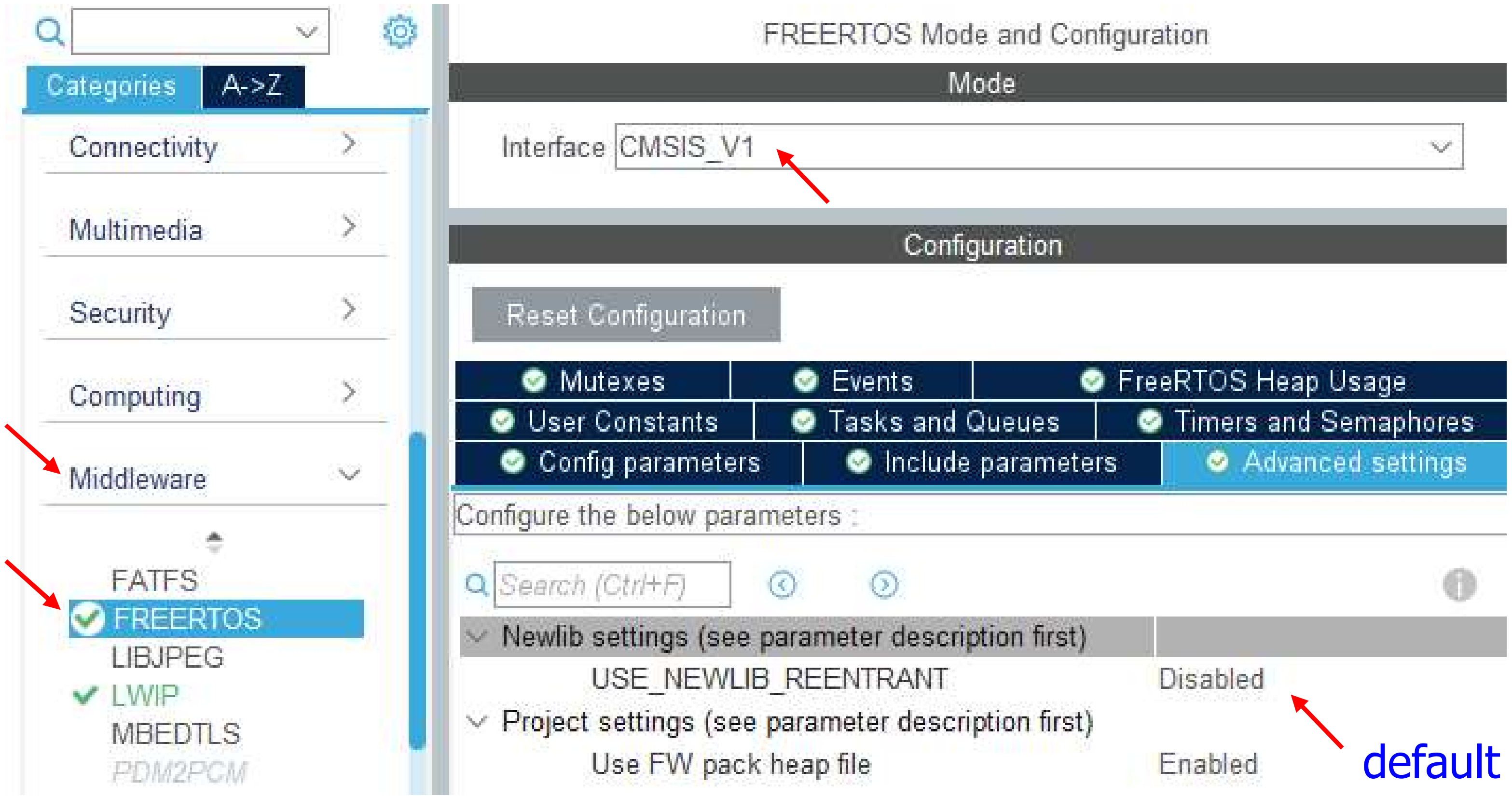


Warning: Code Generation

WARNINGS:

- The USE_NEWLIB_REENTRANT must be set in order to make sure that newlib is fully reentrant.
The option will increase the RAM usage. Enable this option under FreeRTOS > Advanced Settings > USE_NEWLIB_REENTRANT

Do you still want to generate code ?



FREERTOS Mode and Configuration

Interface CMSIS_V1

Newlib settings (see parameter description first)	
USE_NEWLIB_REENTRANT	Disabled
Project settings (see parameter description first)	
Use FW pack heap file	Enabled

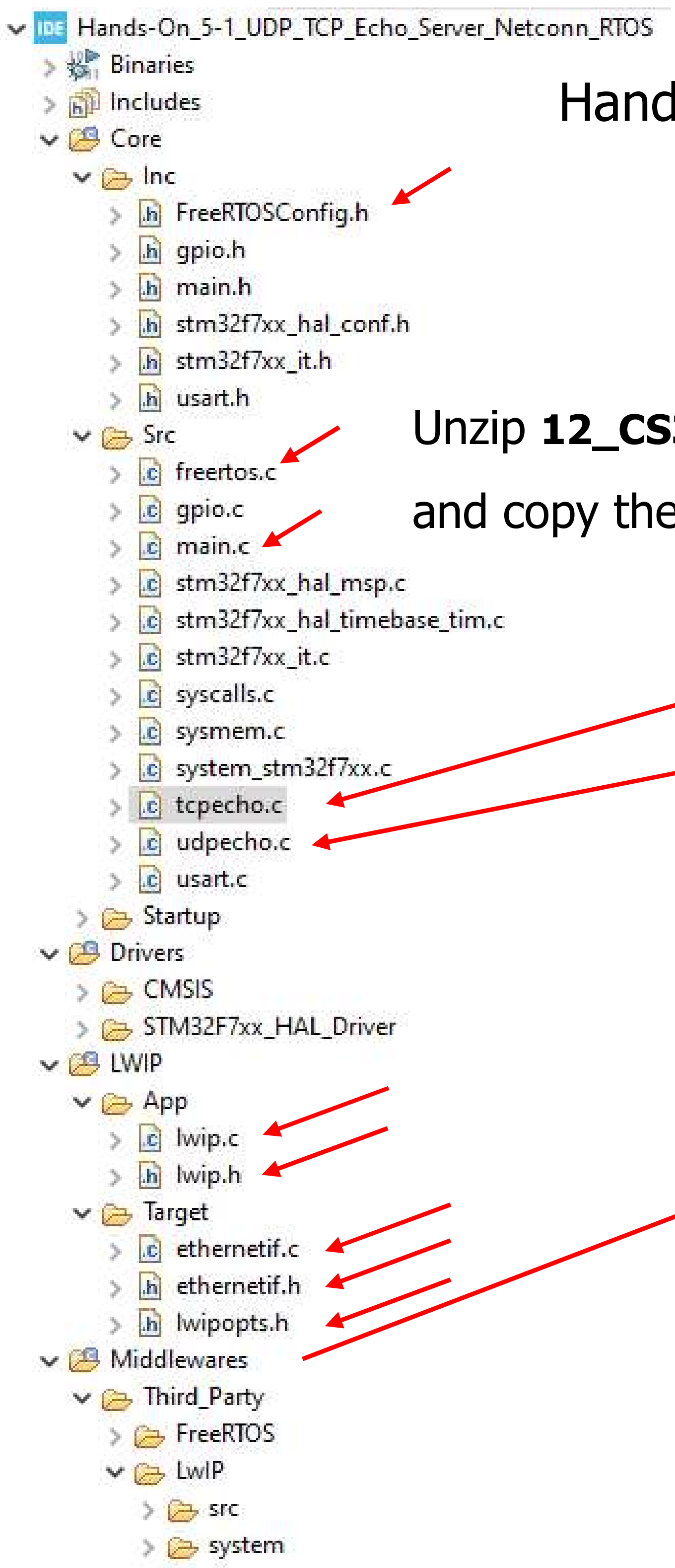
default setting

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

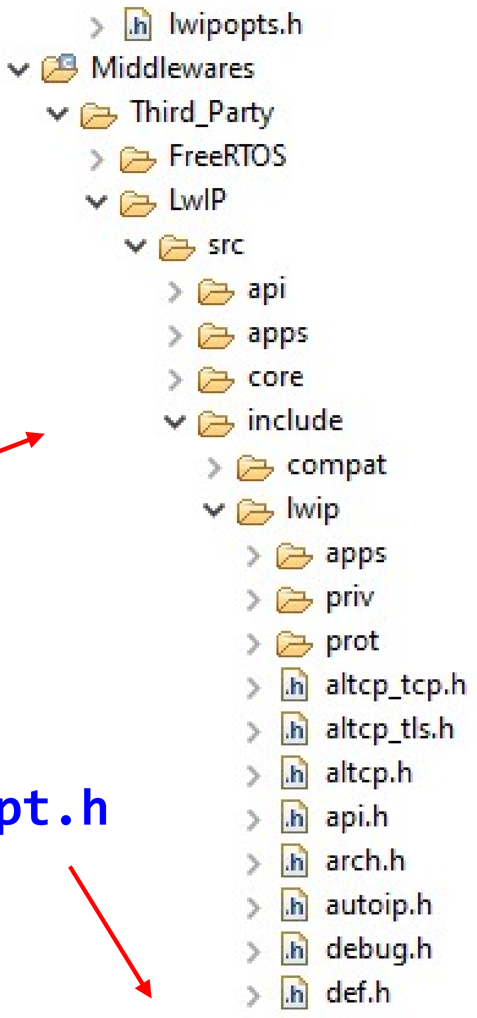
Unzip **12_CS397_Hands-On_5-1_LwIP_udp_tcp_echoserver.zip**, and copy the two files to this project:

tcpecho.c

udpecho.c



```
Hands-On_5-1_UDP_TCP_Echo_Server_Netconn_RTOS
  Binaries
  Includes
  Core
    Inc
      FreeRTOSConfig.h
      gpio.h
      main.h
      stm32f7xx_hal_conf.h
      stm32f7xx_it.h
      usart.h
    Src
      freertos.c
      gpio.c
      main.c
      stm32f7xx_hal_msp.c
      stm32f7xx_hal_timebase_tim.c
      stm32f7xx_it.c
      syscalls.c
      sysmem.c
      system_stm32f7xx.c
      tcpecho.c
      udpecho.c
      usart.c
    Startup
  Drivers
    CMSIS
    STM32F7xx_HAL_Driver
  LWIP
    App
      lwip.c
      lwip.h
    Target
      ethernetif.c
      ethernetif.h
      lwipopts.h
  Middlewares
    Third_Party
      FreeRTOS
      LwIP
        src
        system
```

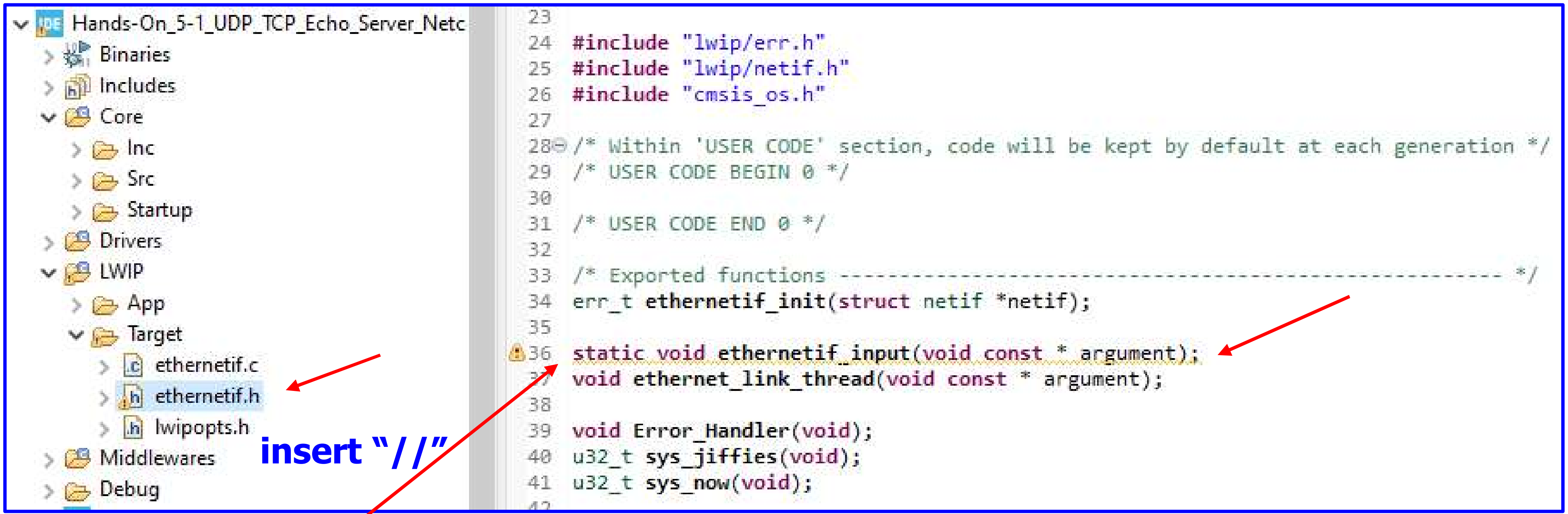


```
      lwipopts.h
  Middlewares
    Third_Party
      FreeRTOS
      LwIP
        src
          api
          apps
          core
          include
            compat
            lwip
              apps
              priv
              prot
              altcp_tcp.h
              altcp_tls.h
              altcp.h
              api.h
              arch.h
              autoip.h
              debug.h
              def.h
```

opt.h

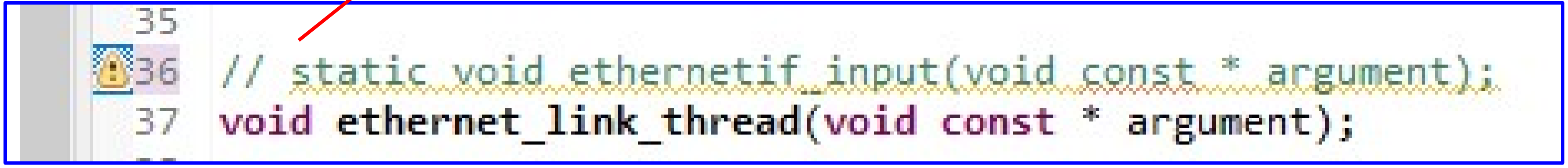
Build warning: Hands-On LwIP UDP TCP Echo Server Netconn RTOS

```
../LWIP/Target/ethernetif.h:36:13: warning: 'ethernetif_input' declared
'static' but never defined [-Wunused-function]
   36 | static void ethernetif_input(void const * argument);
```

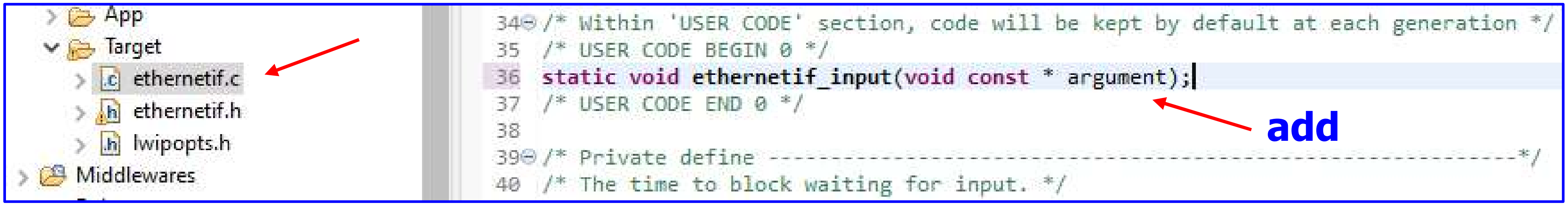


```
#include "lwip/err.h"
#include "lwip/netif.h"
#include "cmsis_os.h"

/* Within 'USER CODE' section, code will be kept by default at each generation */
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/* Exported functions -------------------------------------------------------- */
err_t ethernetif_init(struct netif *netif);

static void ethernetif_input(void const * argument);
void ethernet_link_thread(void const * argument);

void Error_Handler(void);
u32_t sys_jiffies(void);
u32_t sys_now(void);
```

insert "//"



```
// static void ethernetif_input(void const * argument);
void ethernet_link_thread(void const * argument);
```



```
/* Within 'USER CODE' section, code will be kept by default at each generation */
/* USER CODE BEGIN 0 */
static void ethernetif_input(void const * argument);
/* USER CODE END 0 */

/* Private define ------------------------------------------------------------*/
/* The time to block waiting for input. */
```

add

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

Part of the **main.c**

```c
/* Part of the main.c */
/* Includes */
#include "main.h"
#include "cmsis_os.h"
#include "lwip.h"
#include "usart.h"
#include "gpio.h"

/* Private function prototypes */
void SystemClock_Config(void);
void MX_FREERTOS_Init(void);
int main(void)
{
  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();
  /* Configure the system clock */
  SystemClock_Config();

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART3_UART_Init();
  /* Call init function for freertos objects (in freertos.c) */
  MX_FREERTOS_Init();
  /* Start scheduler */
  osKernelStart();

  /* We should never get here as control is now taken by the scheduler */
  /* Infinite loop */
  while (1) { }
}
```

Purpose and Test procedure:

UM1713 User manual

Developing applications on STM32Cube with

LwIP TCP/IP stack

Section 6 Using the LwIP applications

6.1.5 UDP TCP Echo Server based on Netconn RTOS

The **`freertos.c (1/3)`**

Add code

```
/* freertos.c */

/* Includes */
#include "FreeRTOS.h"
#include "task.h"
#include "main.h"
#include "cmsis_os.h"

osThreadId defaultTaskHandle;

/* Private function prototypes */
/* USER CODE BEGIN FunctionPrototypes */
extern void tcpecho_init(void);
extern void udpecho_init(void);
static void ToggleLed1(void const * argument);
/* USER CODE END FunctionPrototypes */

void StartDefaultTask(void const * argument);

extern void MX_LWIP_Init(void);
void MX_FREERTOS_Init(void); /* (MISRA C 2004 rule 8.1) */

/* GetIdleTaskMemory prototype (linked to static allocation support) */
void vApplicationGetIdleTaskMemory( StaticTask_t **ppxIdleTaskTCBBuffer, StackType_t
**ppxIdleTaskStackBuffer, uint32_t *pulIdleTaskStackSize );
```

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

The **`freertos.c (2/3)`**

```
/* USER CODE BEGIN GET_IDLE_TASK_MEMORY */
static StaticTask_t xIdleTaskTCBBuffer;
static StackType_t xIdleStack[configMINIMAL_STACK_SIZE];

void vApplicationGetIdleTaskMemory( StaticTask_t **ppxIdleTaskTCBBuffer, StackType_t
**ppxIdleTaskStackBuffer, uint32_t *pulIdleTaskStackSize )
{
  *ppxIdleTaskTCBBuffer = &xIdleTaskTCBBuffer;
  *ppxIdleTaskStackBuffer = &xIdleStack[0];
  *pulIdleTaskStackSize = configMINIMAL_STACK_SIZE;
  /* place for user code */
}
/* USER CODE END GET_IDLE_TASK_MEMORY */

/* @brief  FreeRTOS initialization */
void MX_FREERTOS_Init(void)
{
  /* Create the thread(s) */
  /* definition and creation of defaultTask */
  osThreadDef(defaultTask, StartDefaultTask, osPriorityNormal, 0, 1024);
  defaultTaskHandle = osThreadCreate(osThread(defaultTask), NULL);

  /* USER CODE BEGIN RTOS_THREADS */
  /* add threads, ... */

  /* Start toogleLed1 task : Toggle LED1  every 500 ms */
  osThreadDef(LED1, ToggleLed1, osPriorityLow, 0, configMINIMAL_STACK_SIZE);
  osThreadCreate (osThread(LED1), NULL);
  /* USER CODE END RTOS_THREADS */
}               // osThreadDef(name, thread, priority, instances, stack_size)
                // NULL or pointer that is passed to the thread function as start argument
```

Add code

The `freertos.c(3/3)` Hands-On LwIP UDP TCP Echo Server Netconn RTOS

```c
void StartDefaultTask(void const * argument)
{
  /* init code for LWIP */
  MX_LWIP_Init();
  /* USER CODE BEGIN StartDefaultTask */

  /* Initialize udp echo server */
  udpecho_init();
  /* Initialize tcp echo server */
  tcpecho_init();

  /* Infinite loop */
  for(;;)
  {
      osDelay(500);
      HAL_GPIO_TogglePin(GPIOB, LD2_Pin);
  }
  /* USER CODE END StartDefaultTask */
}

/* Private application code */
/* USER CODE BEGIN Application */
/*  @brief  Toggle Led1 task */
static void ToggleLed1(void const * argument)
{
  for( ;; )
  {
    /* toggle LED1 at 500 ms */
    HAL_GPIO_TogglePin(GPIOB, LD1_Pin);
    osDelay(500);
  }
}
/* USER CODE END Application */
```

Add code

Add code

Add code

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

Modify Code in the two files:

udpecho.c

```
/* Private function prototypes */
static void udpecho_thread(void const *arg);
void udpecho_init(void);
static void udpecho_thread(void const *arg)
{ . . . }

 void udpecho_init(void)
{
   //sys_thread_new("udpecho_thread", udpecho_thread, NULL, DEFAULT_THREAD_STACKSIZE, osPriorityNormal);
   osThreadDef(TASK_UDP, udpecho_thread, osPriorityNormal, 0, configMINIMAL_STACK_SIZE * 2);
   osThreadCreate (osThread(TASK_UDP), NULL);
}
```

tcpecho.c

```
/* Private function prototypes */
static void tcpecho_thread(void const *arg);
void tcpecho_init(void);
static void udpecho_thread(void const *arg)
{ . . . }

void tcpecho_init(void)
{
   //sys_thread_new("tcpecho_thread", tcpecho_thread, NULL, DEFAULT_THREAD_STACKSIZE, osPriorityNormal);
   osThreadDef(TASK_TCP, tcpecho_thread, osPriorityNormal, 0, configMINIMAL_STACK_SIZE * 2);
   osThreadCreate (osThread(TASK_TCP), NULL);
}
```

The **FreeRTOSConfig.h**

```
/* FreeRTOSConfig.h */

/* Ensure definitions are only used by the compiler, and not by the assembler. */
#if defined(__ICCARM__) || defined(__CC_ARM) || defined(__GNUC__)
  #include <stdint.h>
  extern uint32_t SystemCoreClock;
#endif
#define configENABLE_FPU                         0
#define configENABLE_MPU                         0

#define configUSE_PREEMPTION                     1
#define configSUPPORT_STATIC_ALLOCATION          1
#define configSUPPORT_DYNAMIC_ALLOCATION         1
#define configUSE_IDLE_HOOK                      0
#define configUSE_TICK_HOOK                      0
#define configCPU_CLOCK_HZ                       ( SystemCoreClock )
#define configTICK_RATE_HZ                       ((TickType_t)1000)
#define configMAX_PRIORITIES                     ( 7 )
#define configMINIMAL_STACK_SIZE                 ((uint16_t)1024)
#define configTOTAL_HEAP_SIZE                    ((size_t)63488)
#define configMAX_TASK_NAME_LEN                  ( 16 )
#define configUSE_16_BIT_TICKS                   0
#define configUSE_MUTEXES                        1
#define configQUEUE_REGISTRY_SIZE                8
#define configUSE_PORT_OPTIMISED_TASK_SELECTION  1
/* USER CODE BEGIN MESSAGE_BUFFER_LENGTH_TYPE */
/* Defaults to size_t for backward compatibility, but can be changed
   if lengths will always be less than the number of bytes in a size_t. */
#define configMESSAGE_BUFFER_LENGTH_TYPE         size_t
/* USER CODE END MESSAGE_BUFFER_LENGTH_TYPE */
 . . .
```

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

Add to **lwipots.h** to enable Netconn API

```
/* USER CODE BEGIN 1 */

/* LWIP_NETCONN==1: Enable Netconn API (require to use api_lib.c)  */

#define LWIP_NETCONN      1

/* USER CODE END 1 */
```

Extra, added to remind this option is enabled.

Defined in **opt.h and refer STM32CubeMX LWIP setup**

```
*/
/**
 * @defgroup lwip_opts_netconn Netconn
 * @ingroup lwip_opts_threadsafe_apis
 * @{
 */
/**
 * LWIP_NETCONN==1: Enable Netconn API (require
to use api_lib.c)
 */
#if !defined LWIP_NETCONN || defined __DOXYGEN__
#define LWIP_NETCONN    1
#endif
```

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

```
/* Lwip.c */
/* Variables Initialization */
struct netif gnetif;
ip4_addr_t ipaddr;
ip4_addr_t netmask;
ip4_addr_t gw;
uint8_t IP_ADDRESS[4];
uint8_t NETMASK_ADDRESS[4];
uint8_t GATEWAY_ADDRESS[4];
/* USER CODE BEGIN 2 */
/* USER CODE END 2 */
/* 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;
  /* Initilialize the LwIP stack with RTOS */
```

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;
```

(STM32F767ZI Board)

- IP address
- Netmask address
- Gateway address

Ensure that these settings are correct in STM32CubeMX

Testing of the Application – LwIP UDP TCP Echo Server based on Netconn RTOS

This application provides the echo service for both UDP and TCP protocols.

To test this UDP 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 either:

```
C:\CS397>echotool IP_address /p udp /r 7 /l 7 /n 15 /t 2 /d Testing LwIP UDP echo server
```

or

```
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 udp or /p tcp is the protocol (UDP or TCP 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)

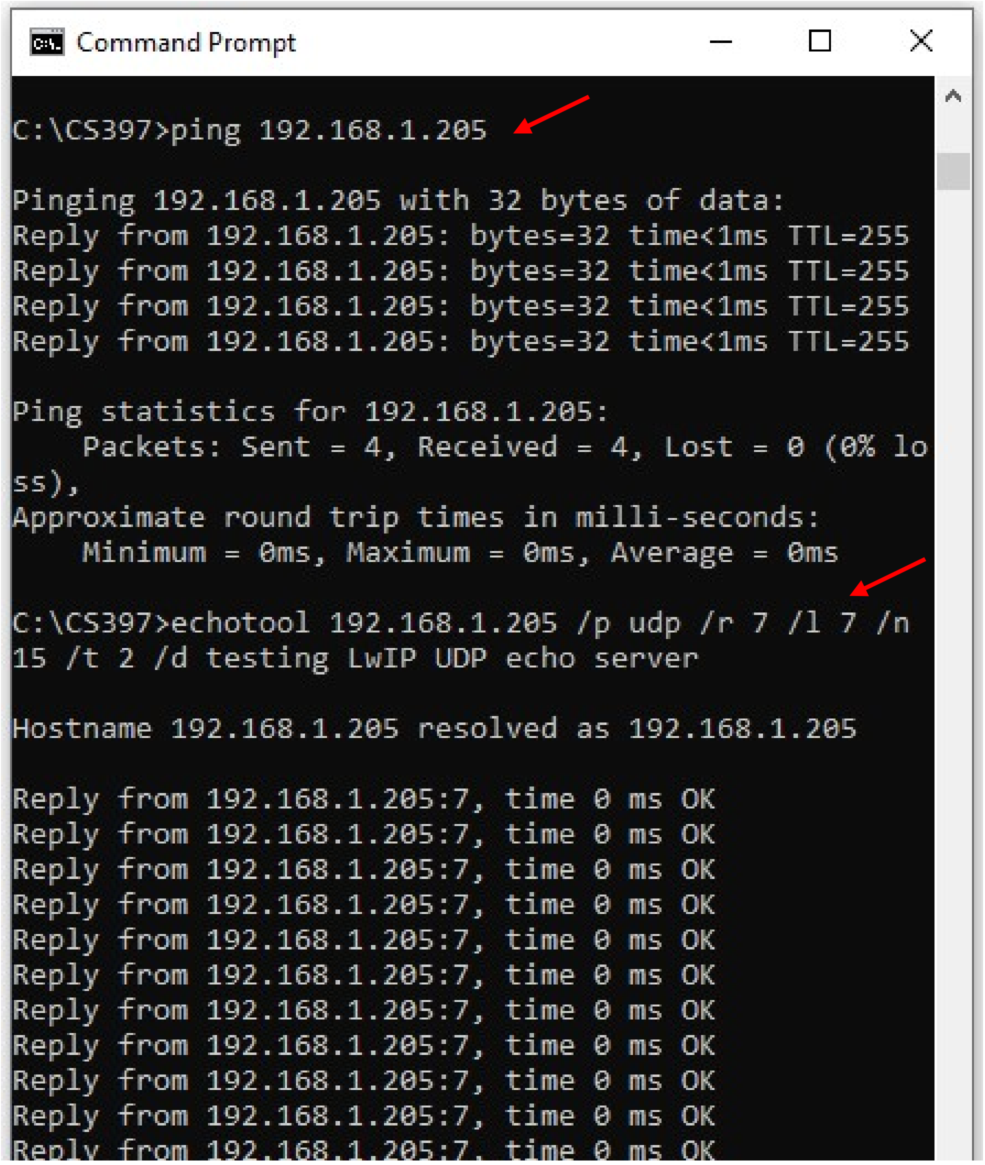
– /d is the message to be sent for echo (for example, “Testing LwIP UDP echo server”)

Note that **reset** the MCU board after a new program is loaded.

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

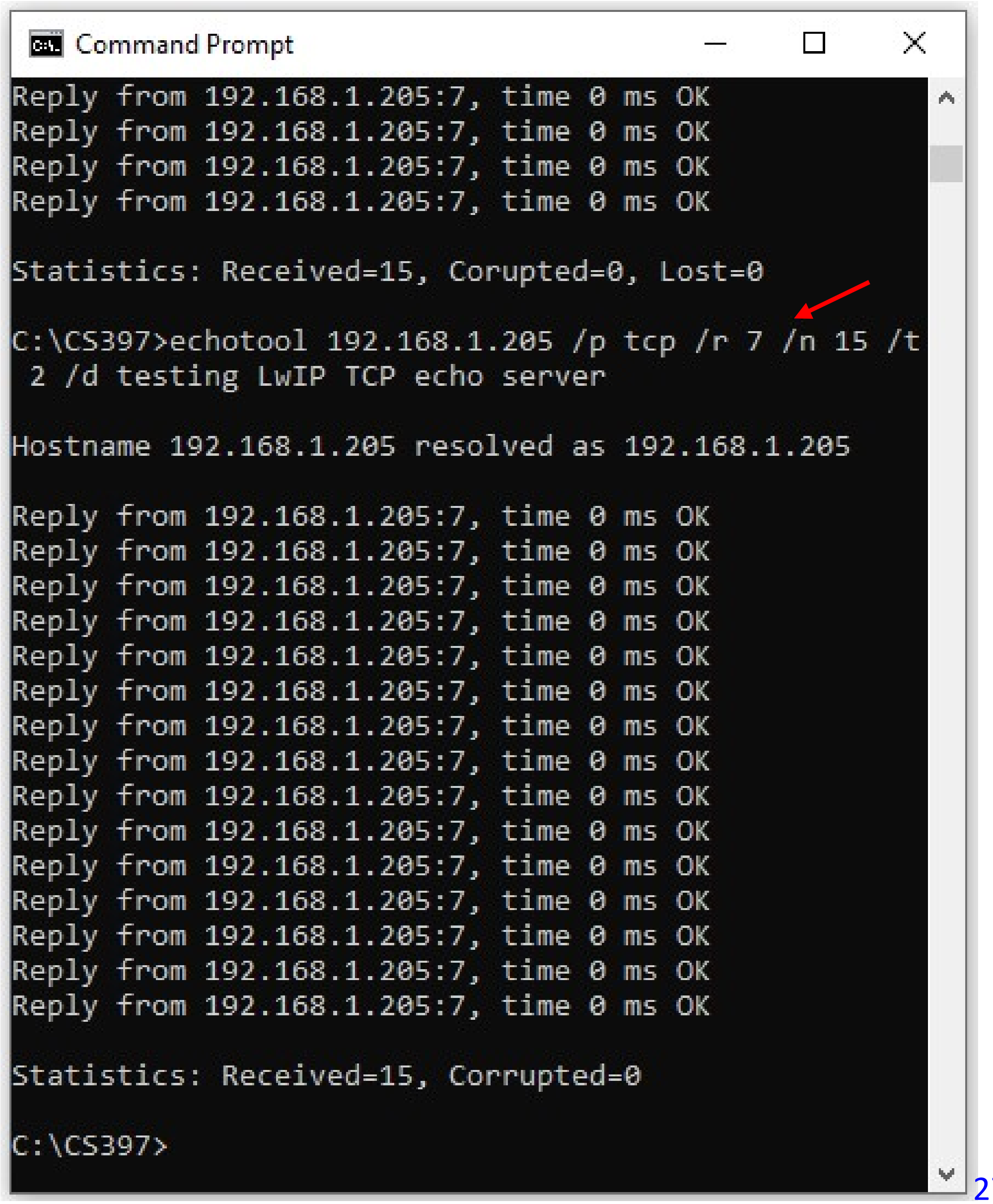
Testing:

1. Create another application with the dynamic address allocation "DHCP" for testing.
2. Figures below show the example of command strings and module responses.



Command Prompt

```
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% lo
ss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

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 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
```

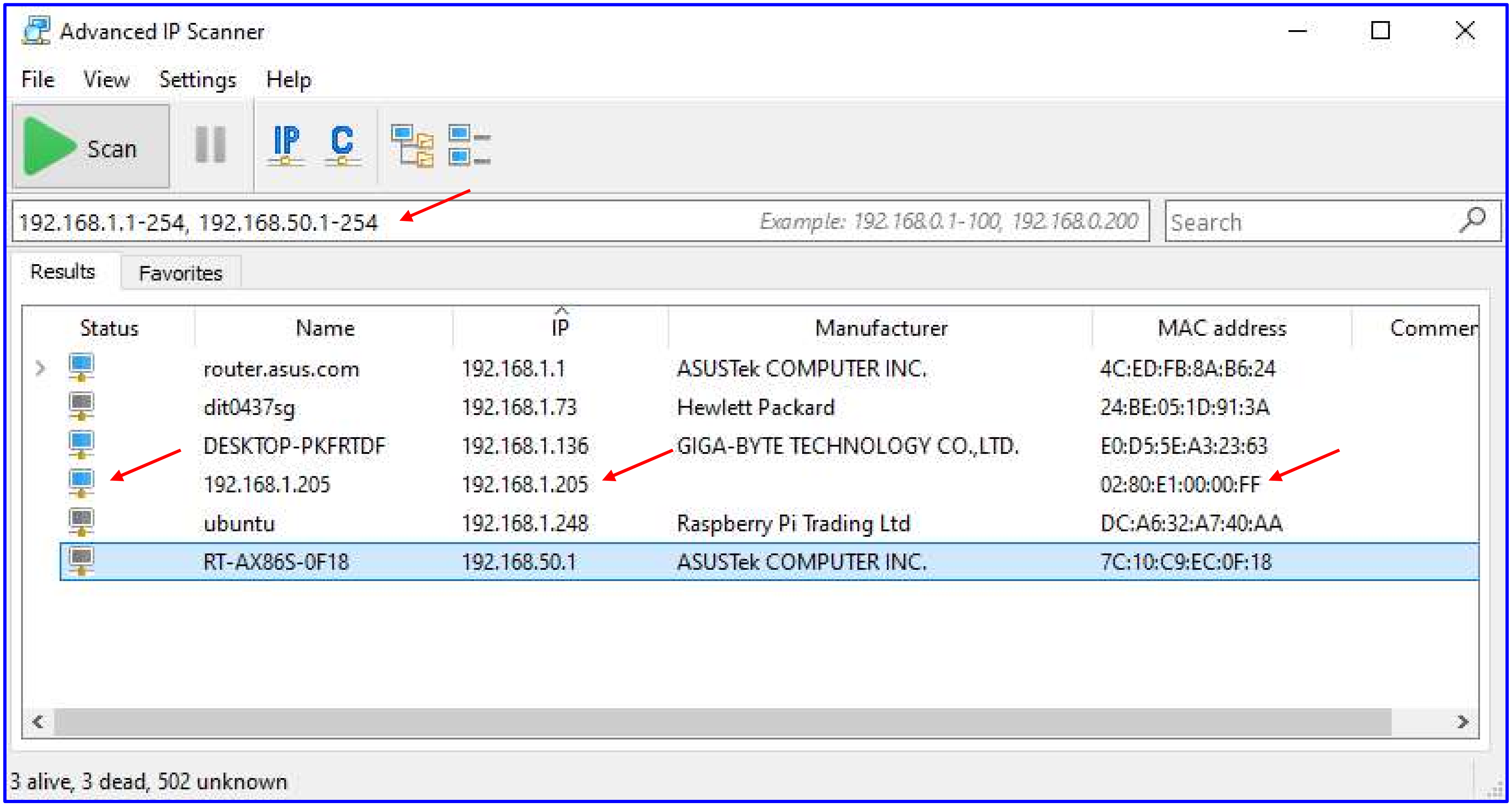


Command Prompt

```
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>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 0 ms OK
Reply from 192.168.1.205:7, time 0 ms OK

Statistics: Received=15, Corrupted=0

C:\CS397>
```

Hands-On LwIP UDP TCP Echo Server Netconn RTOS

- **Advanced IP Scanner –** Scan a network to obtain IP addresses
  - https://www.advanced-ip-scanner.com/
  - **Advanced_IP_Scanner_2.5.4594.1.exe**



192.168.1.1-254, 192.168.50.1-254

Status	Name	IP	Manufacturer	MAC address	Commen
	router.asus.com	192.168.1.1	ASUSTek COMPUTER INC.	4C:ED:FB:8A:B6:24	
	dit0437sg	192.168.1.73	Hewlett Packard	24:BE:05:1D:91:3A	
	DESKTOP-PKFRTDF	192.168.1.136	GIGA-BYTE TECHNOLOGY CO.,LTD.	E0:D5:5E:A3:23:63	
	192.168.1.205	192.168.1.205		02:80:E1:00:00:FF	
	ubuntu	192.168.1.248	Raspberry Pi Trading Ltd	DC:A6:32:A7:40:AA	
	RT-AX86S-0F18	192.168.50.1	ASUSTek COMPUTER INC.	7C:10:C9:EC:0F:18	

3 alive, 3 dead, 502 unknown

- End -