

main.c

* Copyright (C) 2009 - 2014 Xilinx, Inc. All rights reserved.

```
#include <stdio.h>

#include "xparameters.h"

#include "netif/xadapter.h"

#include "platform.h"
#include "platform_config.h"
#if defined (__arm__) || defined(__aarch64__)
#include "xil_printf.h"
#endif

#include "lwip/tcp.h"
#include "xil_cache.h"

#if LWIP_DHCP==1
#include "lwip/dhcp.h"
#endif

/* defined by each RAW mode application */
void print_app_header();
int start_application();
int transfer_data();
void tcp_fasttmr(void);
void tcp_slowtmr(void);

/* missing declaration in lwIP */
void lwip_init();

#if LWIP_DHCP==1
extern volatile int dhcp_timeoutcnt;
err_t dhcp_start(struct netif *netif);
#endif

extern volatile int TcpFastTmrFlag;
extern volatile int TcpSlowTmrFlag;
static struct netif server_netif;
struct netif *echo_netif;

void
print_ip(char *msg, struct ip_addr *ip)
{
    print(msg);
    xil_printf("%d.%d.%d.%d\n\r", ip4_addr1(ip), ip4_addr2(ip),
              ip4_addr3(ip), ip4_addr4(ip));
}
```

Page 1

main.c

```
void
print_ip_settings(struct ip_addr *ip, struct ip_addr *mask, struct ip_addr *gw)
{
    print_ip("Board IP: ", ip);
    print_ip("Netmask : ", mask);
    print_ip("Gateway : ", gw);
}

#if defined (__arm__) && !defined (ARMR5)
#if XPAR_GIGE_PCS_PMA_SGMII_CORE_PRESENT == 1 ||
XPAR_GIGE_PCS_PMA_1000BASEX_CORE_PRESENT == 1
int ProgramSi5324(void);
int ProgramSfpPhy(void);
#endif
#endif

#ifdef XPS_BOARD_ZCU102
#ifdef XPAR_XIICPS_0_DEVICE_ID
int IicPhyReset(void);
#endif
#endif

int main()
{
    struct ip_addr ipaddr, netmask, gw;

    /* the mac address of the board. this should be unique per board */
    unsigned char mac_ethernet_address[] =
    { 0x00, 0x0a, 0x35, 0x00, 0x01, 0x02 };

    echo_netif = &server_netif;
#if defined (__arm__) && !defined (ARMR5)
#if XPAR_GIGE_PCS_PMA_SGMII_CORE_PRESENT == 1 ||
XPAR_GIGE_PCS_PMA_1000BASEX_CORE_PRESENT == 1
    ProgramSi5324();
    ProgramSfpPhy();
#endif
#endif

    /* Define this board specific macro in order perform PHY reset on ZCU102 */
#ifdef XPS_BOARD_ZCU102
    IicPhyReset();
#endif

    init_platform();
}
```

main.c

```
#if LWIP_DHCP==1
    ipaddr.addr = 0;
    gw.addr = 0;
    netmask.addr = 0;
#else
    /* initliaze IP addresses to be used */
    IP4_ADDR(&ipaddr, 192, 168, 1, 10);
    IP4_ADDR(&netmask, 255, 255, 255, 0);
    IP4_ADDR(&gw, 192, 168, 1, 1);
#endif
print_app_header();

lwip_init();

/* Add network interface to the netif_list, and set it as default */
if (!xemac_add(echo_netif, &ipaddr, &netmask,
              &gw, mac_ethernet_address,
              PLATFORM_EMAC_BASEADDR)) {
    xil_printf("Error adding N/W interface\n\r");
    return -1;
}
netif_set_default(echo_netif);

/* now enable interrupts */
platform_enable_interrupts();

/* specify that the network if is up */
netif_set_up(echo_netif);

#if (LWIP_DHCP==1)
/* Create a new DHCP client for this interface.
 * Note: you must call dhcp_fine_tmr() and dhcp_coarse_tmr() at
 * the predefined regular intervals after starting the client.
 */
dhcp_start(echo_netif);
dhcp_timeoutcntr = 24;

while(((echo_netif->ip_addr.addr) == 0) && (dhcp_timeoutcntr > 0))
    xemacif_input(echo_netif);

if (dhcp_timeoutcntr <= 0) {
    if ((echo_netif->ip_addr.addr) == 0) {
        xil_printf("DHCP Timeout\r\n");
        xil_printf("Configuring default IP of 192.168.1.10\r\n");
        IP4_ADDR(&(echo_netif->ip_addr), 192, 168, 1, 10);
        IP4_ADDR(&(echo_netif->netmask), 255, 255, 255, 0);
        IP4_ADDR(&(echo_netif->gw), 192, 168, 1, 1);
    }
}
```

main.c

```
}

ipaddr.addr = echo_netif->ip_addr.addr;
gw.addr = echo_netif->gw.addr;
netmask.addr = echo_netif->netmask.addr;
#endif

print_ip_settings(&ipaddr, &netmask, &gw);

/* start the application (web server, rxtest, txtest, etc..) */
start_application();

/* receive and process packets */
while (1) {
    if (TcpFastTmrFlag) {
        tcp_fasttmr();
        TcpFastTmrFlag = 0;
    }
    if (TcpSlowTmrFlag) {
        tcp_slowtmr();
        TcpSlowTmrFlag = 0;
    }
    xemacif_input(echo_netif);
    transfer_data();
}

/* never reached */
cleanup_platform();

return 0;
}
```