

```

#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include "PCIE.h" // This library belongs to the driver setup

#define PCIE_USER_BAR          PCIE_BAR0 // Defines the PCIE bus bar
#define PCIE_IO_LED_ADDR      0x00     // Address of the LED's
#define PCIE_IO_BUTTON_ADDR   0x20     // Address of the Buttons

typedef enum{
    MENU_BUTTON_LED = 0,
    MENU_QUIT = 99
}MENU_ID;

void UI_ShowMenu(void){
    printf("=====\r\n");
    printf("[%d]: Button/LED control\r\n", MENU_BUTTON_LED);
    printf("[%d]: Quit\r\n", MENU_QUIT);
    printf("Please input your selection:");
}

int UI_UserSelect(void){
    int nSel;
    scanf("%d",&nSel);
    return nSel;
}

BOOL TEST_BUTTON_LED(PCIE_HANDLE hPCIE){
    BOOL bPass1 = TRUE;
    BOOL bPass2;
    DWORD Mask;

    while(Mask>0){
        bPass1 = PCIE_Read32(hPCIE, PCIE_USER_BAR, PCIE_IO_BUTTON_ADDR,&Mask);
        if (bPass1){
            bPass2 = PCIE_Write32(hPCIE, PCIE_USER_BAR, PCIE_IO_LED_ADDR,Mask);
        }
        else
            printf("Failed to read button status\r\n");
    }
    return bPass2;
}

```

```

int main(void)
{
    void *lib_handle;
    PCIE_HANDLE hPCIE;
    BOOL bQuit = FALSE;
    int nSel;

    printf("== Exercise for VIP Module 3 on Embedded Systems ==\r\n");

    lib_handle = PCIE_Load();
    if (!lib_handle){
        printf("PCIE_Load failed!\r\n");
        return 0;
    }

    hPCIE = PCIE_Open(0,0,0);
    if (!hPCIE){
        printf("PCIE_Open failed\r\n");
    }else{
        while(!bQuit){
            UI_ShowMenu();
            nSel = UI_UserSelect();
            switch(nSel){
                case MENU_BUTTON_LED:
                    TEST_BUTTON_LED(hPCIE);
                    break;
                case MENU_QUIT:
                    bQuit = TRUE;
                    printf("Bye!\r\n");
                    break;
                default:
                    printf("Invalid selection\r\n");
            } // switch

        } // while

        PCIE_Close(hPCIE);
    }
    PCIE_Unload(lib_handle);
    return 0;
}

```