

```
/*
 *
 * Xilinx, Inc.
 * XILINX IS PROVIDING THIS DESIGN, CODE, OR INFORMATION "AS IS" AS A
 * COURTESY TO YOU. BY PROVIDING THIS DESIGN, CODE, OR INFORMATION AS
 * ONE POSSIBLE IMPLEMENTATION OF THIS FEATURE, APPLICATION OR
 * STANDARD, XILINX IS MAKING NO REPRESENTATION THAT THIS IMPLEMENTATION
 * IS FREE FROM ANY CLAIMS OF INFRINGEMENT, AND YOU ARE RESPONSIBLE
 * FOR OBTAINING ANY RIGHTS YOU MAY REQUIRE FOR YOUR IMPLEMENTATION
 * XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO
 * THE ADEQUACY OF THE IMPLEMENTATION, INCLUDING BUT NOT LIMITED TO
 * ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE
 * FROM CLAIMS OF INFRINGEMENT, IMPLIED WARRANTIES OF MERCHANTABILITY
 * AND FITNESS FOR A PARTICULAR PURPOSE.
 */

/*
 * Xilinx EDK 12.4 EDK_MS4.81d
 *
 * This file is a sample test application
 *
 * This application is intended to test and/or illustrate some
 * functionality of your system. The contents of this file may
 * vary depending on the IP in your system and may use existing
 * IP driver functions. These drivers will be generated in your
 * XPS project when you run the "Generate Libraries" menu item
 * in XPS.
 *
 * Your XPS project directory is at:
 * C:\Users\jimp\class\codesign\Spartan3E\labs\EDK_INTERRUPT_DEMO_TUT\XPS\
 */

// Located in: microblaze_0/include/xparameters.h
#include "xparameters.h"

#include "xil_cache.h"

#include "stdio.h"

#include "xintc.h"
#include "intc_header.h"
#include "xbasic_types.h"
#include "xgpio.h"
#include "gpio_header.h"
#include "gpio_intr_header.h"
#include "uartlite_header.h"

#define GPIO_CHANNEL1 1

//=====

int main (void) {

    static XIntc intc;

    /*
     * Enable and initialize cache
     */
    #if XPAR_MICROBLAZE_0_USE_ICACHE
        Xil_ICacheInvalidate();
        Xil_ICacheEnable();
    #endif

    #if XPAR_MICROBLAZE_0_USE_DCACHE
        Xil_DCacheInvalidate();
        Xil_DCacheEnable();
    #endif

    static XGpio Push_Buttons_3Bit_Gpio;
    print("-- Entering main() --\r\n");

    {
        int status;

        print("\r\n Running IntcSelfTestExample() for xps_intc_0...\r\n");
    }
}
```

```
status = IntcSelfTestExample(XPAR_XPS_INTC_0_DEVICE_ID);

if (status == 0) {
    print("IntcSelfTestExample PASSED\r\n");
}
else {
    print("IntcSelfTestExample FAILED\r\n");
}
}

{
    int Status;

    Status = IntcInterruptSetup(&intc, XPAR_XPS_INTC_0_DEVICE_ID);
    if (Status == 0) {
        print("Intc Interrupt Setup PASSED\r\n");
    }
    else {
        print("Intc Interrupt Setup FAILED\r\n");
    }
}

{
    u32 status;

    print("\r\nRunning GpioOutputExample() for LEDs_8Bit...\r\n");
    status = GpioOutputExample(XPAR_LEDS_8BIT_DEVICE_ID,8);

    if (status == 0) {
        print("GpioOutputExample PASSED.\r\n");
    }
    else {
        print("GpioOutputExample FAILED.\r\n");
    }
}

{
    u32 status;

    print("\r\nRunning GpioInputExample() for Push_Buttons_3Bit...\r\n");
    u32 DataRead;

    status = GpioInputExample(XPAR_PUSH_BUTTONS_3BIT_DEVICE_ID, &DataRead);

    if (status == 0) {
        xil_printf("GpioInputExample PASSED. Read data:0x%X\r\n", DataRead);
    }
    else {
        print("GpioInputExample FAILED.\r\n");
    }
}

{
    int Status;

    u32 DataRead;

    print(" Press button to Generate Interrupt\r\n");

    Status = GpioIntrExample(&intc, &Push_Buttons_3Bit_Gpio, \
        XPAR_PUSH_BUTTONS_3BIT_DEVICE_ID, \
        XPAR_XPS_INTC_0_PUSH_BUTTONS_3BIT_IP2INTC_IRPT_INTR, \
        GPIO_CHANNEL1, &DataRead);

    if (Status == 0){
        if(DataRead == 0)
            print("No button pressed. \r\n");
        else
            print("Gpio Interrupt Test PASSED. \r\n");
    }
    else {
        print("Gpio Interrupt Test FAILED.\r\n");
    }
}
```

```
}

/*
 * Peripheral SelfTest will not be run for RS232_PORT
 * because it has been selected as the STDOUT device
 */

{
    int status;

    print("\r\nRunning UartLiteSelfTestExample() for mdm_0...\r\n");
    status = UartLiteSelfTestExample(XPAR_MDM_0_DEVICE_ID);
    if (status == 0) {
        print("UartLiteSelfTestExample PASSED\r\n");
    }
    else {
        print("UartLiteSelfTestExample FAILED\r\n");
    }
}

/*
 * Disable cache and reinitialize it so that other
 * applications can be run with no problems
 */
#if XPAR_MICROBLAZE_0_USE_DCACHE
    Xil_DCacheDisable();
    Xil_DCacheInvalidate();
#endif

#if XPAR_MICROBLAZE_0_USE_ICACHE
    Xil_ICacheDisable();
    Xil_ICacheInvalidate();
#endif

print("-- Exiting main() --\r\n");
return 0;
}
```