

```

/*
 * LED_test.c
 *
 * Created on: 13 June 2013
 * Author: Ross Elliot
 * Version: 1.2
 */

/*****
 * VERSION HISTORY
 *****/
 * v1.2 - 13 February 2015
 * Modified for Zybo Development Board ~ DN
 *
 * v1.1 - 27 January 2014
 * GPIO_DEVICE_ID definition updated to reflect new naming conventions in Vivado 2013.3
 * onwards.
 *
 * v1.0 - 13 June 2013
 * First version created.
 *****/

/*****
 * This file contains an example of using the GPIO driver to provide communication between
 * the Zynq Processing System (PS) and the AXI GPIO block implemented in the Zynq Programmable
 * Logic (PL). The AXI GPIO is connected to the LEDs on the Zybo.
 *
 * The provided code demonstrates how to use the GPIO driver to write to the memory mapped AXI
 * GPIO block, which in turn controls the LEDs.
 *****/

/* Include Files */
#include "xparameters.h"
#include "xgpio.h"
#include "xstatus.h"
#include "xil_printf.h"

/* Definitions */

```

```

#define GPIO_DEVICE_ID  XPAR_AXI_GPIO_0_DEVICE_ID  /* GPIO device that LEDs are connected to */
#define LED 0x9 /* Initial LED value - X00X */
#define LED_DELAY 10000000 /* Software delay length */
#define LED_CHANNEL 1 /* GPIO port for LEDs */
#define printf xil_printf /* smaller, optimised printf */

XGpio Gpio; /* GPIO Device driver instance */

int LEDOutputExample(void)
{
    volatile int Delay;
    int Status;
    int led = LED; /* Hold current LED value. Initialise to LED definition */

    /* GPIO driver initialisation */
    Status = XGpio_Initialize(&Gpio, GPIO_DEVICE_ID);
    if (Status != XST_SUCCESS) {
        return XST_FAILURE;
    }

    /*Set the direction for the LEDs to output. */
    XGpio_SetDataDirection(&Gpio, LED_CHANNEL, 0x0);

    /* Loop forever blinking the LED. */
    while (1) {
        /* Write output to the LEDs. */
        XGpio_DiscreteWrite(&Gpio, LED_CHANNEL, led);

        /* Flip LEDs. */
        led = ~led;

        /* Wait a small amount of time so that the LED blinking is visible. */
        for (Delay = 0; Delay < LED_DELAY; Delay++);
    }

    return XST_SUCCESS; /* Should be unreachable */
}

```

```
/* Main function. */
int main(void){

    int Status;

    /* Execute the LED output. */
    Status = LEDOutputExample();
    if (Status != XST_SUCCESS) {
        xil_printf("GPIO output to the LEDs failed!\r\n");
    }

    return 0;
}
```