Application Note No. 001
TITLE

Richard Thavot Joe Doe Marco Mattavelli

Ecole Polytechnique Fédérale de Lausanne
Multimedia Group Laboratory
SCI-STI-MM

March 25, 2011

1.1 Introduction

1.1.1 Purpose

```
/*********************************************************************
 Interrupt Service Routin for Timer 0.
*********************************************************************/
// LED4
static action ADI_INT_HANDLER(Timer0_ISR) {

   // See if this is a timer 0 event, by calling a function to
   // read corresponding bit (16) in the the SIC_ISR register

   if (adi_int_SICInterruptAsserted(ADI_INT_TIMER0) == ADI_INT_RESULT_NOT_ASSERTED)

   // This return value tells the interrupt manager to process the next
   // ISR in the chain for this IVG, because we haven't yet serviced the
   // peripheral that interrupted this time

      return (ADI_INT_RESULT_NOT_PROCESSED);

   // clear timer 0 interupt
   adi_tmr_GPControl(ADI_TMR_GP_TIMER_0, ADI_TMR_GP_CMD_CLEAR_INTERRUPT, NULL);

   // toggle the specified LED
   // ON|ON|ON
   if(button0 && button1 && button2){
      ezToggleLED(EZ_FIRST_LED);          // LED4,5,6 BLINK
   }

   //***CODE SNIPPED****//

   return (ADI_INT_RESULT_PROCESSED);

}

   //***CODE SNIPPED****//

/*************************************************************************
  Function: Init_Timers

  Set up timers for PWM mode and enale them.
*************************************************************************/

void InitTimers(void)
{
   //Setting up command table for Timer 0
   ADI_TMR_GP_CMD_VALUE_PAIR Timer0ConfigurationTable [] = {
      { ADI_TMR_GP_CMD_SET_TIMER_MODE,         (void *)0x01          },
      { ADI_TMR_GP_CMD_SET_COUNT_METHOD,          (void *)TRUE          },
      { ADI_TMR_GP_CMD_SET_INTERRUPT_ENABLE,     (void *)TRUE          },
      { ADI_TMR_GP_CMD_SET_OUTPUT_PAD_DISABLE,   (void *)TRUE          },
      //***CODE SNIPPED****//
      { ADI_TMR_GP_CMD_SET_WIDTH,              (void *)0x00400000      },
      { ADI_TMR_GP_CMD_END,                    NULL                },
   };

      //***CODE SNIPPED****//

}

/*********************************************************************
*  Function:    main
*********************************************************************/
```

```
61 void main(void) {

63     u32 ResponseCount;
       void *pExitCriticalArg;
65     u32 i; //loop variable

67
       // initialize the EZ-Kit
69     ezInit(1);

71     // initialize the flag manager because the LEDs and buttons connect via flags
       // Since callbacks are not being used memory does not to be given to the service
73     ezErrorCheck(adi_flag_Init(NULL, 0, &ResponseCount, NULL));

75     //***CODE SNIPPED****//

77     InitTimers();

79     while (1) {
       //***CODE SNIPPED****//
81        }

83     } // END WHILE
   } // END MAIN
```