Repeatedly Turn On the LED for 3s and Then Off for 1s

1. Here is one way to do it. We know that the ACLK increments the peripheral approximately every 25µs. Therefore, to count for 1 seconds, we would effectively need to count:

   1 second / 25µs = 40,000 times

   The program counts to 40,000 three times while the red LED is on. Then, it turns off the red LED and counts to 40,000 one time before turning the red LED back on and repeating.

   We have put the program on the next page to keep it all on the same sheet of paper.

```
#include <msp430.h>

#define     RED_LED         0x0001              // P1.0 is the red LED
#define     DEVELOPMENT     0x5A80              // Stop the watchdog timer
#define     ENABLE_PINS     0xFFFE              // Required to use inputs and outputs
#define     ACLK            0x0100              // Timer_A ACLK source
#define     UP              0x0010              // Timer_A UP mode
#define     TAIFG           0x0001              // Used to look at Timer A Interrupt FlaG

main()
{
    unsigned char intervals=0;                  // Count number of 40,000 counts

    WDTCTL  = DEVELOPMENT;                      // Stop the watchdog timer
    PM5CTL0 = ENABLE_PINS;                      // Enable inputs and outputs

    TA0CCR0 = 40000;                            // We will count up from 0 to 40,000
    TA0CTL  = ACLK | UP;                        // Use ACLK, for UP mode

    P1DIR   = RED_LED;                          // Set red LED as an output
    P1OUT   = RED_LED;                          // Turn red LED on

    while(1)
    {
        if(TA0CTL & TAIFG)                      // If timer has counted to 40,000
        {
              TA0CTL = TA0CTL & (~TAIFG);       //    Count again
              intervals = intervals + 1;        //    Update number of 40,000 counts

              if (P1OUT & RED_LED)              //    If the red LED is on
              {
                  if (intervals == 3)           //       Have 3*40,000 counts gone by?
                  {
                      P1OUT = 0x00;             //          If yes, turn off LED
                      intervals = 0;            //          Clear number of 40,000 counts
                  }
              }
              else                              //    Else, red LED is off
              {
                  P1OUT = RED_LED;              //       Turn on the red LED
                  intervals = 0;                //       Clear number of 40,000 counts
              }

        }

    }// end while(1)

}// end main()
```

All tutorials and software examples included herewith are intended solely for educational purposes. The material is provided in an "as is" condition. Any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for particular purposes are disclaimed.

The software examples are self-contained low-level programs that typically demonstrate a single peripheral function or device feature in a highly concise manner. Therefore, the code may rely on the device's power-on default register values and settings such as the clock configuration and care must be taken when combining code from several examples to avoid potential side effects. Additionally, the tutorials and software examples should not be considered for use in life support devices or systems or mission critical devices or systems.

In no event shall the owner or contributors to the tutorials and software be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.