```c
// Timing C/ASM Mix Example
// Jason Losh

//-----------------------------------------------------------------------------
// Hardware Target
//-----------------------------------------------------------------------------

// Target Platform: EK-TM4C123GXL Evaluation Board
// Target uC:       TM4C123GH6PM
// System Clock:    40 MHz

// Hardware configuration:
// Red LED:
//   PF1 drives an NPN transistor that powers the red LED

//-----------------------------------------------------------------------------
// Device includes, defines, and assembler directives
//-----------------------------------------------------------------------------

#include <stdint.h>
#include <stdbool.h>
#include "tm4c123gh6pm.h"

#define RED_LED      (*((volatile uint32_t *)(0x42000000 + (0x400253FC-0x40000000)*32 + 1*4)))

//-----------------------------------------------------------------------------
// Subroutines
//-----------------------------------------------------------------------------

// Initialize Hardware
void initHw()
{
    // Configure HW to work with 16 MHz XTAL, PLL enabled, system clock of 40 MHz
    SYSCTL_RCC_R = SYSCTL_RCC_XTAL_16MHZ | SYSCTL_RCC_OSCSRC_MAIN | SYSCTL_RCC_USESYSDIV | (4 << SYSCTL_RCC_SYSDIV_S);

    // Set GPIO ports to use APB (not needed since default configuration -- for clarity)
    SYSCTL_GPIOHBCTL_R = 0;

    // Enable GPIO port F peripherals
    SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF;

    // Configure LED and pushbutton pins
    GPIO_PORTF_DIR_R |= 0x02;  // make bit 1 an outputs
    GPIO_PORTF_DR2R_R |= 0x02; // set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTF_DEN_R |= 0x02;  // enable LED
}

// Approximate busy waiting (in units of microseconds), given a 40 MHz system clock
void waitMicrosecond(uint32_t us)
{
    __asm("WMS_LOOP0:   MOV  R1, #6");         // 1
    __asm("WMS_LOOP1:   SUB  R1, #1");         // 6
    __asm("             CBZ  R1, WMS_DONE1");  // 5+1*3
    __asm("             NOP");                 // 5
    __asm("             NOP");                  // 5
    __asm("             B    WMS_LOOP1");       // 5*2 (speculative, so P=1)
    __asm("WMS_DONE1:   SUB  R0, #1");          // 1
    __asm("             CBZ  R0, WMS_DONE0");   // 1
    __asm("             NOP");                  // 1
    __asm("             B    WMS_LOOP0");       // 1*2 (speculative, so P=1)
    __asm("WMS_DONE0:");                        // ---
                                                // 40 clocks/us + error
}

//-----------------------------------------------------------------------------
// Main
//-----------------------------------------------------------------------------

int main(void)
{
    // Initialize hardware
    initHw();

    // Toggle red LED every second
    while(1)
    {
      RED_LED ^= 1;
      waitMicrosecond(1000000);
    }
}

```