```c
/*
 * CE2812 - 021
 * Winter 2016
 * Lab 3 - LCD API
 * Name: Yahui Liang
 * Created: 12/17/2016
 */

/* All header files */
#include "delay_api.h"
#include "reg_struct.h"

/* Private method prototypes */
static void wait_delay();
static void reset_regs();

static volatile SYSTICK* systick = (SYSTICK*) SYSTICK_BASE;

/**
 * Delays the number of milliseconds.
 * Args:
 * theDelay: the number of ms which needs to be delayed.
 */
void delay_ms(uint32_t theDelay) {
	reset_regs();
	systick -> STK_LOAD = theDelay * (F_CPU / 8000);
	wait_delay(systick);
}

/**
 * Delays the number of microseconds.
 * Args:
 * theDelay: the number of us which needs to be delayed.
 */
void delay_us(uint32_t theDelay) {
	reset_regs(systick);
	systick -> STK_LOAD = theDelay * (F_CPU / 8000000);
	wait_delay(systick);
}

/**
 * Resets the CTRL register and the VAL register to be zero.
 */
static void reset_regs() {
	systick -> STK_VAL = 0;
	systick -> STK_CTRL = 0;
}

/**
 * Waits the count flag to be set. If the flag is set, the method disables
 * the timer.
 */
static void wait_delay() {
	systick -> STK_CTRL = 1; // enable the clock.
	while (!(systick -> STK_CTRL & (1 << 16))) {
			// nothing to do.
		}
```

```
        systick -> STK_CTRL = 0;
}
```