```
/*******************************************************************************
 * Name:    lab2_lib.c
 * Description: STM32 peripherals initialization and functions
 * Version: V1.00
 * Author: Li Pan
 *
 * This software is supplied "AS IS" without warranties of any kind.
 *
 *
 *-----------------------------------------------------------------------------
 * History:
 *          V1.00 Initial Version
 *          V1.1 reformatted (kjn)
 ******************************************************************************/
 #include "stm32f10x.h"
 #include "CLOCK.h"



/*************************************************
* This function will alternate the LEDs on the VLDiscovery board
**************************************************/
void clockInit(void)
{
    uint32_t temp = 0x00;
    //If you hover over the RCC you can go to the definition and then
    //see it is a structure of all the RCC registers.  Then you can
    //simply assign a value.
   // RCC->CFGR = 0x07050002;     // Output PLL/2 as MCO,
                                // PLLMUL X3, PREDIV1 is PLL input
      RCC->CFGR = 0x00050002;

    RCC->CR =  0x01010081;      // Turn on PLL, HSE, HSI

    while (temp != 0x02000000)  // Wait for the PLL to stabilize
    {
        temp = RCC->CR & 0x02000000; //Check to see if the PLL lock bit is set
    }

    //Enable peripheral clocks for various ports and subsystems
    //Bit 4: Port C Bit3: Port B Bit 2: Port A
    RCC->APB2ENR |=  RCC_APB2ENR_IOPAEN ;

    // Write a 0xB ( 1011b ) into the configuration and mode bits for PA8 (AFIO)
    GPIOA->CRH |= GPIO_CRH_CNF8_1 | GPIO_CRH_MODE8_1 | GPIO_CRH_MODE8_0 ;
    GPIOA->CRH &= ~GPIO_CRH_CNF8_0 ;



}

/*
* Name:         void delay()
* Paramaters:   32 bit delay value, ( a value of 6000
                gives approximately 1 mS of delay)
* Description:  This function creates a delay
*/
void delay(uint32_t count)
{
    int i=0;
    for(i=0; i< count; ++i)
    {
    }
}

// SWITCH AND LED I/O FUNCTIONS

void led_IO_init (void)
{
    //Enable peripheral clocks for various ports and subsystems
    //Bit 4: Port C Bit3: Port B Bit 2: Port A
    RCC->APB2ENR |=  RCC_APB2ENR_IOPCEN | RCC_APB2ENR_IOPBEN
        | RCC_APB2ENR_IOPAEN ;

    //Set the config and mode bits for Port C bit 9 and 8 so they will
    // be push-pull outputs (up to 50 MHz)
    GPIOC->CRH |= GPIO_CRH_MODE9 | GPIO_CRH_MODE8 ;
    GPIOC->CRH &= ~GPIO_CRH_CNF9 & ~GPIO_CRH_CNF8 ;



}

/**************************************************
* This function will alternate the LEDs on the VLDiscovery board
**************************************************/
void led_wigwag (void)
{
    int i = 1200000;

    // Read the USER button and select the LED pattern according to its state
    if ((GPIOA->IDR & GPIO_IDR_IDR0) == 0)
    {
        GPIOC->BSRR = BLUE_ON | GREEN_OFF;
        delay(i);
        GPIOC->BSRR = GREEN_ON | BLUE_OFF;
        delay(i);
    }
    else
    {
        GPIOC->BSRR = BLUE_ON | GREEN_ON;
        delay(i);
        GPIOC->BSRR = GREEN_OFF | BLUE_OFF;
        delay(i);
    }
}
```