```c
/*
	CE2812 Lab 2
	Knight Rider LEDs in C
	Evan Heinrich
	12/10/2021
*/


#include "registers.h"
#include <stdio.h>
#include <stdlib.h>
#include "uart_driver.h"
#include "LED.h"
#include "delay.h"

#define F_CPU 16000000UL
#define MOTION 50

// main
int main(void){
	// Initialize required libraries
	init_usart2(57600,F_CPU);
	delay_Init();
	LED_Init();

	// Infinite loop
	while(1==1) {
		// Starting position
		int light = 1;

		// Shift left loop
		for(int i = 0; i < 10; i++) {
			int shifted = light << i;
			LED_PrintNum(shifted);
			delay_ms(MOTION);
		}

		// Light shifted all the way to the left
		light = 1 << 9;

		// Shift right loop
		for(int i = 0; i < 10; i++) {
			int shifted = light >> i;
			LED_PrintNum(shifted);
			delay_ms(MOTION);
		}
	}

	exit(EXIT_SUCCESS);

	return 0;
}
```

```c
/*
	CE2812
	Delay code
	Evan Heinrich
	Created Lab2
	Note: Uses TIM2 prescaled to 1kHz
*/

#include "registers.h"
#include <inttypes.h>
void delay_Init() {
	// Initialize Variables
	volatile uint32_t* addr = 0;
	volatile uint32_t contents = 0;
	volatile uint32_t mask = 0;

	// Enable TIM2 in APB1ENR
	addr = (uint32_t*)RCC_APB1ENR;				// Set target address
	mask = 1 << 0;						// Set mask
	contents = *addr | mask;  // Read & modify
	*addr = contents;				// Write

	// Set TIM2 CR configurations
	addr = (uint32_t*)TIM2_CR1;	// Set target address
	mask = (1<<3)|(1<<4);			// Set mask for one-pulse and count down
	contents = *addr | mask;  // Read & modify
	*addr = contents;				// Write

	// Set TIM2 prescale (no RMW needed)
	addr = (uint32_t*)TIM2_PSC;	// Set target address
	*addr = 16000;					// Set prescale to 16kHz

	// Prescale fix
	// Forces an event to be generated and then
	// clears it right away which tricks the timer
	// into applying the prescale somehow
	addr = (uint32_t*)TIM2_EGR;
	*addr = 1;
	addr = (uint32_t*)TIM2_SR;
	*addr &= ~(1);

	return;
}
```

```c
void delay_ms(uint32_t delay) {
	// Initialize Variables
	volatile uint32_t* addr = 0;
	volatile uint32_t contents = 0;
	volatile uint32_t mask = 0;

	// Assert counting is not enabled
	addr = (uint32_t*)TIM2_CR1;	// Set target address
	mask = ~(1<<0);				// Set mask to clear CEN
	contents = *addr & mask;  // Read & modify
	*addr = contents;			// Write

	// Write the desired count (no RMW necessary)
	addr = (uint32_t*)TIM2_CNT;	// Set target address
	*addr = delay;				// Write desired delay

	// Enable count
	addr = (uint32_t*)TIM2_CR1;	// Set target address
	mask = (1<<0);				// Set mask
	contents = *addr | mask;  // Read & modify
	*addr = contents;			// Write

	// Busy loop
	addr = (uint32_t*)TIM2_CR1;
	mask = (1<<0);
	contents = *addr & mask;

	while(contents != 0) {
		contents = *addr & mask;
	}

	return;
}
```

```c
/*
	CE2812
	LED Driver
	Evan Heinrich
	Created Lab2
	Displays a 10-bit number on the LEDs on the MSOE Devboard
*/

#include "registers.h"
#include <inttypes.h>
void LED_Init() {
	// Initialize Variables
	volatile uint32_t* addr = 0;
	volatile uint32_t mask = 0;

	// Enable GPIOB in AHB1
	addr = (uint32_t*) RCC_AHB1ENR;  // Set target address
	mask = 1<<1;        // Set mask to enable GPIOB
	*addr |= mask;             // Slick read modify write

	// Set pins PB5-PB15 (skip PB11) as outputs
	addr = (uint32_t*) GPIOB_MODER;  // Set target address
	mask = 0x55155400;                      // Mask to set appropriate pins
	*addr |= mask;                                  // RMW

	return;
}

void LED_PrintNum(uint32_t num) {
	// Initialize Variables
	volatile uint32_t* addr = 0;

	// Split the argument because PB11 is skipped
	uint32_t temp = num & (1023);

	// Upper 4 bits
	uint32_t upper = temp & 960;

	// Lower 6 bits
	uint32_t lower = temp & 63;

	// Value to be displayed with PB11 skipped
	uint32_t output = ((upper << 1) | lower) << 5;

	// Clear LEDs via BSRR
	addr = (uint32_t*)GPIOB_BSRR;
	*addr = 0xFFFF0000;

	*addr = output;
	return;
}
```

```c
/*
	CE2812
	Register addresses for our NODE F446RE boards
	Evan Heinrich
	NOTE: Running list of all registers used, updated as new
	peripherals are used
*/

#ifndef REG_LIST_ALIVE
#define REG_LIST_ALIVE 1

#define RCC_APB1ENR 0x40023840
#define RCC_AHB1ENR 0x40023830
#define TIM2_CR1 0x40000000
#define TIM2_PSC 0x40000028
#define TIM2_CNT 0x40000024
#define TIM2_EGR 0x40000014
#define TIM2_SR 0x40000010
#define TIM2_ARR 0x4000002C
#define GPIOB_MODER 0x40020400
#define GPIOB_ODR 0x40020414
#define GPIOB_BSRR 0x40020418

#endif


/*
	CE2812
	LED Header
	Evan Heinrich
	Created Lab2
*/

#ifndef LED_DRIVER_ALIVE
#define LED_DRIVER_ALIVE 1

void LED_Init();

void LED_PrintNum(uint32_t);

#endif



/*
	CE2812
	Delay header
	Evan Heinrich
	Created Lab2
*/

#ifndef DELAY_DRIVER_ALIVE
#define DELAY_DRIVER_ALIVE 1

void delay_Init();

void delay_ms(uint32_t);

#endif
```