```c
/*******************************************************************************

  * @file    main.c
  * @author  John Bretz
  * @version V1.0
  * @brief   Entry point and main logic.
  ******************************************************************************
*/

#include <stdio.h>
#include <stdlib.h>
#include "embedded.h"

#define delay 100

int main(void) {

    LED_INIT();

    int number = 1;

    while (1) {
        for (int i = 0; i < 9; i++) {
            light_LED(number);
            delay_ms(delay);
            number = number << 1;
        }
        for (int i = 0; i < 9; i++) {
            light_LED(number);
            delay_ms(delay);
            number = number >> 1;
        }
    }
}
```

```c
/**
  ******************************************************************************
  * @file    led.c
  * @author  John Bretz
  * @version V1.0
  * @brief   LED api.
  ******************************************************************************
*/

#define RCC_AHB1ENR (int *) 0x40023830
#define GPIOB_MODER (int *) 0x40020400
#define GPIOB_ODR (int *) 0x40020414

#define GPIOB_ENABLE 0x2

void LED_INIT(void) {
    // turn on gpiob
    *RCC_AHB1ENR |= GPIOB_ENABLE;

    // set pb5-10, pb12-15 to output
    *GPIOB_MODER |=  0x55155400;
    *GPIOB_MODER &= ~0xAA2AA800;
}

void light_LED(int number) {
    *GPIOB_ODR &= ~0xF7E0; // turn off all leds

    *GPIOB_ODR |= (number & 0x3F)  << 5; // bottom 6 bits
    *GPIOB_ODR |= (number & 0x3C0) << 6; // top 4 bits
}
```

```c
/**
  ******************************************************************************
  * @file    timing.c
  * @author  John Bretz
  * @version V1.0
  * @brief   Timing api.
  ******************************************************************************
*/


#define STK_BASE (int *) 0xE000E010
#define STK_CLK_SOURCE (int *) 0xE000E012
#define STK_LOAD (int *) 0xE000E014

#define COUNT_FLAG 1<<16
#define freq 16000000UL

void delay_ms(int delay) {

    *STK_BASE = 0;                      // disable clock
    *STK_CLK_SOURCE = 0;                // use system clock
    *STK_LOAD = delay * (freq / 8000); // set delay
    *STK_BASE = 1;                      // enable the clock.

    while (!(*STK_BASE & (1 << 16))) {
        // busy wait
    }

    *STK_BASE = 0; // disable clock
}
```

```c
/**
 ******************************************************************************
 * @file embedded.h
 * @author John Bretz
 * @version V1.0
 * @brief Function prototypes
 ******************************************************************************
 */

/*
 * Sets up the LED array for use
 */
void LED_INIT(void);

/*
 * A busy wait that will last delay ms
 */
void delay_ms(int delay);

/*
 * Displays a ten bit binary number on the LED array
 */
void light_LED(int number);
```