Task 2:

Submitted By: Sagar More

Write proper structure implementation for allRCC registers and bit field implementation AHBENR1 register bits and structure and bit field you have to do for IDR and ODR and MODE register and GIPO regarding

#Main.c

```c
/*******************************************************************************
 * @file           : main.c
 * @author         : Sagar More
 * @brief          : Class 3 - Task Led and Button using structure
 * @board          : STM32F446RE
 * ******************************************************************************/

#include "HeaderTask2.h"

int main(void){
    /* Enable GPIOA clock */
    RCC->AHB1ENR.AHB1ENR0 = 0x01;

    /* Enable GPIOC Clock */
    RCC->AHB1ENR.AHB1ENR2 = 0x01;

    /* Set PA5 as Output (01) */
    GPIOA->MODER.MODER5 = 0x01;

    /* Set PC13 as Input (00) */
    GPIOC->MODER.MODER13 = 0x00;

    while(1){
        if (GPIOC->IDR.IDR13 == 0){
            GPIOA->ODR.ODR5 = 1;   // LED ON
        }
        else{
            GPIOA->ODR.ODR5 = 0;   // LED OFF
        }
    }
}
```

#Header.h

```
/*
 * HeaderTask2.h
 *
 *  Created on: Oct 11, 2025
 *      Author: Sagar More
 */
#include <stdint.h>

#ifndef HEADERTASK2_H_
#define HEADERTASK2_H_

#define __IO volatile

/* ------------------------------------------------------------------
 * Peripheral base addresses
 * ------------------------------------------------------------------
*/
#define PERIPH_BASE           0x40000000UL
#define AHB1PERIPH_BASE       (PERIPH_BASE + 0x00020000UL)
#define AHB2PERIPH_BASE       (PERIPH_BASE + 0x10000000UL)
#define APB1PERIPH_BASE       (PERIPH_BASE + 0x00000000UL)
#define APB2PERIPH_BASE       (PERIPH_BASE + 0x00010000UL)

/* ------------------------------------------------------------------
 * Specific peripheral base addresses
 * ------------------------------------------------------------------
*/
#define RCC_BASE              (AHB1PERIPH_BASE + 0x3800UL)
#define GPIOA_BASE            (AHB1PERIPH_BASE + 0x0000UL)
#define GPIOB_BASE            (AHB1PERIPH_BASE + 0x0400UL)
#define GPIOC_BASE            (AHB1PERIPH_BASE + 0x0800UL)
#define GPIOD_BASE            (AHB1PERIPH_BASE + 0x0C00UL)
#define GPIOE_BASE            (AHB1PERIPH_BASE + 0x1000UL)


/*--------------------------------------------------------------
 * GPIO IDR Register structure
 * -------------------------------------------------------------*/
typedef struct
{
    uint32_t IDR0  : 1;
    uint32_t IDR1  : 1;
    uint32_t IDR2  : 1;
    uint32_t IDR3  : 1;
    uint32_t IDR4  : 1;
    uint32_t IDR5  : 1;
    uint32_t IDR6  : 1;
    uint32_t IDR7  : 1;
    uint32_t IDR8  : 1;
    uint32_t IDR9  : 1;
    uint32_t IDR10 : 1;
    uint32_t IDR11 : 1;
    uint32_t IDR12 : 1;
    uint32_t IDR13 : 1;
    uint32_t IDR14 : 1;
    uint32_t IDR15 : 1;
```

```
    uint32_t RESERVED : 16;
} GPIOx_IDR;


/*--------------------------------------------
 * GPIO ODR Register structure
 * -------------------------------------------*/
typedef struct
{
    uint32_t ODR0  : 1;
    uint32_t ODR1  : 1;
    uint32_t ODR2  : 1;
    uint32_t ODR3  : 1;
    uint32_t ODR4  : 1;
    uint32_t ODR5  : 1;
    uint32_t ODR6  : 1;
    uint32_t ODR7  : 1;
    uint32_t ODR8  : 1;
    uint32_t ODR9  : 1;
    uint32_t ODR10 : 1;
    uint32_t ODR11 : 1;
    uint32_t ODR12 : 1;
    uint32_t ODR13 : 1;
    uint32_t ODR14 : 1;
    uint32_t ODR15 : 1;
    uint32_t RESERVED : 16;
} GPIOx_ODR;



/* ---------------------------------------------------------------------------
 * RCC_AHB1ENR register structure
 * ---------------------------------------------------------------------------
*/
typedef struct
{
    uint32_t AHB1ENR0    : 1;  // GPIOA Enable Bit
    uint32_t AHB1ENR1    : 1;  // GPIOB Enable Bit
    uint32_t AHB1ENR2    : 1;  // GPIOC Enable Bit
    uint32_t AHB1ENR3    : 1;  // GPIOD Enable Bit
    uint32_t AHB1ENR4    : 1;  // GPIOE Enable Bit
    uint32_t AHB1ENR5    : 1;  // GPIOF Enable Bit
    uint32_t AHB1ENR6    : 1;  // GPIOG Enable Bit
    uint32_t AHB1ENR7    : 1;  // GPIOH Enable Bit
    uint32_t RESERVED1   : 4;  // Reserved bits
    uint32_t AHB1ENR12   : 1;  // CRC Enable Bit
    uint32_t RESERVED2   : 5;  // Reserved Bits
    uint32_t AHB1ENR18   : 1;  // BKP SRAMEN Bit
    uint32_t RESERVED3   : 2;  // Reserved Bits
    uint32_t AHB1ENR21   : 1;  // DMA Enable Bit
    uint32_t AHB1ENR22   : 1;  // DMA Enable Bit
    uint32_t RESERVED4   : 6;  // Reserved Bits
    uint32_t AHB1ENR29   : 1;  // OTGHS Enable Bit
    uint32_t AHB1ENR30   : 1;  // OTGHS ULP Enable Bit
    uint32_t RESERVED5   : 1;  // Reserved Bit
} RCC_AHB1ENR;
```

```
/*--------------------------------------------------------------------
 * GPIO MODER Register
 *--------------------------------------------------------------------
*/
typedef struct
{
    uint32_t MODER0  : 2;
    uint32_t MODER1  : 2;
    uint32_t MODER2  : 2;
    uint32_t MODER3  : 2;
    uint32_t MODER4  : 2;
    uint32_t MODER5  : 2;
    uint32_t MODER6  : 2;
    uint32_t MODER7  : 2;
    uint32_t MODER8  : 2;
    uint32_t MODER9  : 2;
    uint32_t MODER10 : 2;
    uint32_t MODER11 : 2;
    uint32_t MODER12 : 2;
    uint32_t MODER13 : 2;
    uint32_t MODER14 : 2;
    uint32_t MODER15 : 2;
} GPIOx_MODER;


/*--------------------------------------------------------------------
 * GPIO Register structure
 *--------------------------------------------------------------------
-*/

typedef struct
{
    __IO GPIOx_MODER MODER;   /* GPIO port mode register*/
    __IO uint32_t    OTYPER;  /* GPIO port output type register*/
    __IO uint32_t    OSPEEDR; /* GPIO port output speed register*/
    __IO uint32_t    PUPDR;   /* GPIO port pull-up/pull-down register*/
    __IO GPIOx_IDR   IDR;     /* GPIO port input data register*/
    __IO GPIOx_ODR   ODR;     /* GPIO port output data register*/
    __IO uint32_t    BSRR;    /* GPIO port bit set/reset register*/
    __IO uint32_t    LCKR;    /* GPIO port config lock register*/
    __IO uint32_t    AFR[2];  /* GPIO alternate function registers,*/
} GPIO_TypeDef;


/* ------------------------------------------------------------------
 * RCC register structure
 * ------------------------------------------------------------------
*/
typedef struct
{
    __IO uint32_t CR;
    __IO uint32_t PLLCFGR;
    __IO uint32_t CFGR;
    __IO uint32_t CIR;
    __IO uint32_t AHB1RSTR;
    __IO uint32_t AHB2RSTR;
    __IO uint32_t AHB3RSTR;
    __IO uint32_t RESERVED0;
```

```
    __IO uint32_t APB1RSTR;
    __IO uint32_t APB2RSTR;
    __IO uint32_t RESERVED1[2];
    __IO RCC_AHB1ENR AHB1ENR;     // We are using AHB1ENR
    __IO uint32_t AHB2ENR;
    __IO uint32_t AHB3ENR;
    __IO uint32_t RESERVED2;
    __IO uint32_t APB1ENR;
    __IO uint32_t APB2ENR;
    __IO uint32_t RESERVED3[2];
    __IO uint32_t AHB1LPENR;
    __IO uint32_t AHB2LPENR;
    __IO uint32_t AHB3LPENR;
    __IO uint32_t RESERVED4;
    __IO uint32_t APB1LPENR;
    __IO uint32_t APB2LPENR;
    __IO uint32_t RESERVED5[2];
    __IO uint32_t BDCR;
    __IO uint32_t CSR;
    __IO uint32_t RESERVED6[2];
    __IO uint32_t SSCGR;
    __IO uint32_t PLLI2SCFGR;
    __IO uint32_t PLLSAICFGR;
    __IO uint32_t DCKCFGR;
    __IO uint32_t CKGATENR;
    __IO uint32_t DCKCFGR2;
} RCC_TypeDef;


/*-------------------------------------------------
 * Creating instances
 * -----------------------------------------------*/
#define GPIOA ((GPIO_TypeDef *) GPIOA_BASE)
#define GPIOB ((GPIO_TypeDef *) GPIOB_BASE)
#define GPIOC ((GPIO_TypeDef *) GPIOC_BASE)
#define GPIOD ((GPIO_TypeDef *) GPIOD_BASE)
#define GPIOE ((GPIO_TypeDef *) GPIOE_BASE)

#define RCC   ((RCC_TypeDef  *) RCC_BASE)


#endif /* HEADERTASK2_H_ */
```