

<<IERG3810>>

<<Microcontroller and Embedded Systems Laboratory>>

Report on Experiment <<5>>

<<Timer>>

Group: 19

Member: Chan Kai Yin (CUID: 1155124983)

Ip Tsun Yu (CUID: 1155144668)

Submission Date: 23-03-2022

Disclaimer

I declare that the assignment here submitted is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website <http://www.cuhk.edu.hk/policy/academichonesty/>

Jesse

Chan Kai Yin

23-03-2022

Signature

Name

Date

Derek

Ip Tsun Yu

23-03-2022

Signature

Name

Date

I. OBJECTIVES

- To study the Timer setting of Cortex-M3.
- To compare the response time of read-modify-write and direct modification of a register.
- To compare the overhead and the switching time of interrupt handler and subroutines.
- To study SYSTICK setting of Cortex-M3.
- To study the generation of a PWM signal.
- To study remapping of cortex-M3 Alternate Function.
- To learn the use of a timer and its multiple-channel outputs to drive a tri-color LED

II. DATA ANALYSIS

Experiment 5.1

The LED is flashing in 1 Hz:

```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
#include "FONT.H"
#include "SevenSegments.h"
#include "CFONT.H"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
    u32 i;
    for(i = 0; i < count; i++);
}

void IERG3810_key2_ExtInit(){
    // EXTI-2
    RCC->APB2ENR |= 1 << 6;
    GPIOE ->CRL &= 0xFFFFF0FF;
    GPIOE ->CRL |= 0x00000800;
    GPIOE ->BSRR = 1 << 2;
    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[0] &= 0xFFFFF0FF;
    AFIO ->EXTICR[0] |= 0x00000400;
```

```

EXTI ->IMR |= 1 << 2;
EXTI ->FTSR |= 1 << 2;
//EXTI ->RTSR |= 1 << 2;

NVIC ->IP[8] = 0x65;
NVIC ->ISER[0] &= ~(1 << 8);
NVIC ->ISER[0] |= (1 << 8);

}

void IERG3810_NVIC_SetPriorityGroup(u8 prigroup){
    u32 temp, temp1;
    temp1 = prigroup & 0x00000007;
    temp1 <<= 8;
    temp = SCB ->AIRCRR;
    temp &= 0x0000F8FF;
    temp |= 0x05FA0000;
    temp |= temp1;
    SCB ->AIRCRR = temp;
}

void EXTI2_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOB ->BRR = 1 << 5;
        Delay(1000000);
        GPIOB ->BSRR = 1 << 5;
        Delay(1000000);
    }
    EXTI ->PR = 1 << 2;
}

void EXTI0_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOE ->BRR = 1 << 5; // on
        Delay(1000000);
        GPIOE ->BSRR = 1 << 5; // off
        Delay(1000000);
    }
    EXTI ->PR = 1 << 0;
}

void IERG3810_keyUP_ExtInit(){
    //PA0, KeyUp press = high, EXTI-0
    RCC->APB2ENR |= 1 << 2;
    GPIOA ->CRL &= 0xFFFFFFFF0;
    GPIOA ->CRL |= 0x00000008;
    GPIOA ->ODR |= 1 << 4;
}

```

```

RCC->APB2ENR |= 0x1;
AFIO ->EXTICR[0] &= 0xFFFFFFF0; //EXTI-0
AFIO ->EXTICR[0] |= 0x00000000; //EXTI-0, 0000: PA[x] pin

EXTI ->IMR |= 0x1; //0001 not mask on line 0
EXTI ->FTSR |= 0x1; // Falling trigger enabled

NVIC ->IP[6] = 0x95; //priority = 0x95
NVIC ->ISER[0] &= ~(1 << 6); //enable IRQ 6 for EXTI0
NVIC ->ISER[0] |= (1 << 6); //IRQ 6
}

void IERG3810_PS2key_ExtInit(){
    // PS2 data : PC10, PS2 CLK: PC11

    RCC->APB2ENR |= 1 << 4;
    GPIOC ->CRH &= 0xFFFF00FF; //PC10, PC11
    GPIOC ->CRH |= 0x00008800; // 1000

    GPIOC ->BSRR = 1 << 11; // set high
    GPIOC ->BSRR = 1 << 10;

    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[2] &= 0xFFFF0FFF; //EXTI11
    AFIO ->EXTICR[2] |= 0x00002000; // 0010: PC11 pin

    EXTI ->IMR |= 1 << 11; // Event request from Line x is not masked
    EXTI ->FTSR |= 1 << 11; // Falling trigger enabled

    NVIC ->IP[40] = 0x65; //priority for IRQ 40
    NVIC ->ISER[1] |= (1 << 8); //enable IRQ 40 for EXTI[15:10]
}

u32 sheep = 0;
u32 timeout = 10000;
u32 ps2key = 0;
u32 tmp = 0;
u32 ps2count = 0;
u8 ps2dataReady = 0;
u8 key_stack[2];

void EXTI15_10_IRQHandler(void){

    if (ps2count > 0 && ps2count < 9){ //1:8 , bit 0:7 data

        tmp = ps2key >>= 1; //right shift 1
        if ((GPIOC->IDR)&(1<<10)){
            tmp |= 0x80;
        }
        ps2key = tmp;
        ps2count++;
    }
}

```

```

    }

    else {
        ps2count++;
    }

    Delay(10);
    EXTI->PR = 1 << 11;
}

void IERG3810_TIM3_Init(u16 arr, u16 psc){

    //Tim3, IRQ#29
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;
    TIM3->PSC = psc;
    TIM3->DIER |= 1 << 0;
    TIM3->CR1 |= 0x01;
    NVIC->IP[29] = 0x45;
    NVIC->ISER[0] |= (1 << 29);
}

void TIM3_IRQHandler(void){
    if(TIM3->SR & 1 << 0){
        GPIOB->ODR ^= 1 << 5;
    }
    TIM3->SR &=~(1 << 0);
}

int main(void){
    IERG3810_LED_Init();
    IERG3810_clock_tree_init();
    IERG3810_NVIC_SetPriorityGroup(5);
    IERG3810_TIM3_Init(4999,7199);
    while(1){
        ;
    }
}

```

Experiment 5.2

The ds0 is flashing in 1Hz and ds1 is flashing in 4Hz.

4Hz calculated with the formula: $72M/7200/1250/2 = 4$

```

#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"

```

```

#include "FONT.H"
#include "SevenSegments.h"
#include "CFONT.H"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
    u32 i;
    for(i = 0; i < count; i++);
}

void IERG3810_key2_ExtInit(){
    // EXTI-2
    RCC->APB2ENR |= 1 << 6;
    GPIOE ->CRL &= 0xFFFFF0FF;
    GPIOE ->CRL |= 0x00000800;
    GPIOE ->BSRR = 1 << 2;
    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[0] &= 0xFFFFF0FF;
    AFIO ->EXTICR[0] |= 0x00000400;
    EXTI ->IMR |= 1 << 2;
    EXTI ->FTSR |= 1 << 2;
    //EXTI ->RTSR |= 1 << 2;

    NVIC ->IP[8] = 0x65;
    NVIC ->ISER[0] &= ~(1 << 8);
    NVIC ->ISER[0] |= (1 << 8);
}

void IERG3810_NVIC_SetPriorityGroup(u8 prigroup){
    u32 temp, temp1;
    temp1 = prigroup & 0x00000007;
    temp1 <<= 8;
    temp = SCB ->AIRCRR;
    temp &= 0x0000F8FF;
    temp |= 0x05FA0000;
    temp |= temp1;
    SCB ->AIRCRR = temp;
}

void EXTI2_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){

```

```

        GPIOB ->BRR = 1 << 5;
        Delay(1000000);
        GPIOB ->BSRR = 1 << 5;
        Delay(1000000);
    }
    EXTI ->PR = 1 << 2;
}

void EXTI0_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOE ->BRR = 1 << 5; // on
        Delay(1000000);
        GPIOE ->BSRR = 1 << 5; // off
        Delay(1000000);
    }
    EXTI ->PR = 1 << 0;
}

void IERG3810_keyUP_ExtInit(){
    //PA0, KeyUp press = high, EXTI-0
    RCC->APB2ENR |= 1 << 2;
    GPIOA ->CRL &= 0xFFFFFFFF0;
    GPIOA ->CRL |= 0x00000008;
    GPIOA ->ODR |= 1 << 4;

    RCC->APB2ENR |= 0x1;
    AFIO ->EXTICR[0] &= 0xFFFFFFFF0; //EXTI-0
    AFIO ->EXTICR[0] |= 0x00000000; //EXTI-0, 0000: PA[x] pin

    EXTI ->IMR |= 0x1; //0001 not mask on line 0
    EXTI ->FTSR |= 0x1; // Falling trigger enabled

    NVIC ->IP[6] = 0x95; //priority = 0x95
    NVIC ->ISER[0] &= ~(1 << 6); //enable IRQ 6 for EXTI0
    NVIC ->ISER[0] |= (1 << 6); //IRQ 6
}

void IERG3810_PS2key_ExtInit(){
    // PS2 data : PC10, PS2 CLK: PC11

    RCC->APB2ENR |= 1 << 4;
    GPIOC ->CRH &= 0xFFFF00FF; //PC10, PC11
    GPIOC ->CRH |= 0x00008800; // 1000

    GPIOC ->BSRR = 1 << 11; // set high

```

```

GPIOC ->BSRR = 1 << 10;

RCC->APB2ENR |= 0x01;
AFIO ->EXTICR[2] &= 0xFFFF0FFF; //EXTI11
AFIO ->EXTICR[2] |= 0x00002000; // 0010: PC11 pin

EXTI ->IMR |= 1 << 11; // Event request from Line x is not masked
EXTI ->FTSR |= 1 << 11; // Falling trigger enabled

NVIC ->IP[40] = 0x65; //priority for IRQ 40
NVIC ->ISER[1] |= (1 << 8); //enable IRQ 40 for EXTI[15:10]
}

u32 sheep = 0;
u32 timeout = 10000;
u32 ps2key = 0;
u32 tmp = 0;
u32 ps2count = 0;
u8 ps2dataReady = 0;
u8 key_stack[2];

void EXTI15_10_IRQHandler(void){

    if (ps2count > 0 && ps2count < 9){ //1:8 , bit 0:7 data

        tmp = ps2key >>= 1; //right shift 1
        if ((GPIOC->IDR)&(1<<10)){
            tmp |= 0x80;
        }
        ps2key = tmp;
        ps2count++;
    }

    else {
        ps2count++;
    }

    Delay(10);
    EXTI->PR = 1 << 11;
}

void IERG3810_TIM3_Init(u16 arr, u16 psc){

    //Tim3, IRQ#29
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;

```



```

    TIM3->PSC = psc;
    TIM3->DIER |= 1 << 0;
    TIM3->CR1 |= 0x01;
    NVIC->IP[29] = 0x45;
    NVIC->ISER[0] |= (1 << 29);
}

void TIM3_IRQHandler(void){
    if(TIM3->SR & 1 << 0){
        GPIOB->ODR ^= 1 << 5;
    }
    TIM3->SR &=~(1 << 0);
}

void IERG3810_TIM4_Init(u16 arr, u16 psc){

    //Tim4, IRQ#30
    RCC->APB1ENR |= 1 << 2;      // TIM4
    TIM4->ARR = arr;
    TIM4->PSC = psc;
    TIM4->DIER |= 1 << 0;
    TIM4->CR1 |= 0x01;
    NVIC->IP[30] = 0x45;
    NVIC->ISER[0] |= (1 << 30);
}

void TIM4_IRQHandler(void){
    if(TIM4->SR & 1 << 0){
        GPIOE->ODR ^= 1 << 5;
    }
    TIM4->SR &=~(1 << 0);
}

int main(void){
    IERG3810_LED_Init();
    IERG3810_clock_tree_init();
    IERG3810_NVIC_SetPriorityGroup(5);
    // arr, psc
    IERG3810_TIM3_Init(4999,7199);
    IERG3810_TIM4_Init(1249,7199);    // 72M/7200/1250/2 = 4

    while(1){
        ;
    }
}

```

Experiment 5.3

```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
#include "FONT.H"
#include "SevenSegments.h"
#include "CFONT.H"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
    u32 i;
    for(i = 0; i < count; i++);
}

void IERG3810_key2_ExtInit(){
    // EXTI-2
    RCC->APB2ENR |= 1 << 6;
    GPIOE ->CRL &= 0xFFFFF0FF;
    GPIOE ->CRL |= 0x00000800;
    GPIOE ->BSRR = 1 << 2;
    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[0] &= 0xFFFFF0FF;
    AFIO ->EXTICR[0] |= 0x00000400;
    EXTI ->IMR |= 1 << 2;
    EXTI ->FTSR |= 1 << 2;
    //EXTI ->RTSR |= 1 << 2;

    NVIC ->IP[8] = 0x65;
    NVIC ->ISER[0] &= ~(1 << 8);
    NVIC ->ISER[0] |= (1 << 8);
}

void IERG3810_NVIC_SetPriorityGroup(u8 prigroup){
    u32 temp, temp1;
    temp1 = prigroup & 0x00000007;
    temp1 <= 8;
    temp = SCB ->AIRCRR;
    temp &= 0x0000F8FF;
    temp |= 0x05FA0000;
    temp |= temp1;
    SCB ->AIRCRR = temp;
}
```

```

void EXTI2_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOB ->BRR = 1 << 5;
        Delay(1000000);
        GPIOB ->BSRR = 1 << 5;
        Delay(1000000);
    }
    EXTI ->PR = 1 << 2;
}

void EXTI0_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOE ->BRR = 1 << 5; // on
        Delay(1000000);
        GPIOE ->BSRR = 1 << 5; // off
        Delay(1000000);
    }
    EXTI ->PR = 1 << 0;
}

void IERG3810_keyUP_ExtInit(){
    //PA0, KeyUp press = high, EXTI-0
    RCC->APB2ENR |= 1 << 2;
    GPIOA ->CRL &= 0xFFFFFFFF0;
    GPIOA ->CRL |= 0x00000008;
    GPIOA ->ODR |= 1 << 4;

    RCC->APB2ENR |= 0x1;
    AFIO ->EXTICR[0] &= 0xFFFFFFFF0; //EXTI-0
    AFIO ->EXTICR[0] |= 0x00000000; //EXTI-0, 0000: PA[x] pin

    EXTI ->IMR |= 0x1; //0001 not mask on line 0
    EXTI ->FTSR |= 0x1; // Falling trigger enabled

    NVIC ->IP[6] = 0x95; //priority = 0x95
    NVIC ->ISER[0] &= ~(1 << 6); //enable IRQ 6 for EXTI0
    NVIC ->ISER[0] |= (1 << 6); //IRQ 6
}

void IERG3810_PS2key_ExtInit(){
    // PS2 data : PC10, PS2 CLK: PC11

    RCC->APB2ENR |= 1 << 4;
    GPIOC ->CRH &= 0xFFFF00FF; //PC10, PC11
    GPIOC ->CRH |= 0x00008800; // 1000

    GPIOC ->BSRR = 1 << 11; // set high
    GPIOC ->BSRR = 1 << 10;
}

```

```

RCC->APB2ENR |= 0x01;
AFIO ->EXTICR[2] &= 0xFFFF0FFF; //EXTI11
AFIO ->EXTICR[2] |= 0x00002000; // 0010: PC11 pin

EXTI ->IMR |= 1 << 11; // Event request from Line x is not masked
EXTI ->FTSR |= 1 << 11; // Falling trigger enabled

NVIC ->IP[40] = 0x65; //priority for IRQ 40
NVIC ->ISER[1] |= (1 << 8); //enable IRQ 40 for EXTI[15:10]
}

u32 sheep = 0;
u32 timeout = 10000;
u32 ps2key = 0;
u32 tmp = 0;
u32 ps2count = 0;
u8 ps2dataReady = 0;
u8 key_stack[2];

void EXTI15_10_IRQHandler(void){

    if (ps2count > 0 && ps2count < 9){ //1:8 , bit 0:7 data

        tmp = ps2key >>= 1; //right shift 1
        if ((GPIOC->IDR)&(1<<10)){
            tmp |= 0x80;
        }
        ps2key = tmp;
        ps2count++;
    }

    else {
        ps2count++;
    }

    Delay(10);
    EXTI->PR = 1 << 11;
}

void IERG3810_TIM3_Init(u16 arr, u16 psc){

    //Tim3, IRQ#29
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;
    TIM3->PSC = psc;
    TIM3->DIER |= 1 << 0;
    TIM3->CR1 |= 0x01;
    NVIC->IP[29] = 0x45;
    NVIC->ISER[0] |= (1 << 29);
}

```

```

}

void TIM3_IRQHandler(void){

    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    /*
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    */
    /*
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    */
    TIM3->SR &= ~(1<<0);
    TIM3->SR &= ~(1<<0);
}

void IERG3810_TIM4_Init(u16 arr, u16 psc){

    //Tim4, IRQ#30
    RCC->APB1ENR |= 1 << 2;          // TIM4
    TIM4->ARR = arr;
    TIM4->PSC = psc;
    TIM4->DIER |= 1 << 0;
    TIM4->CR1 |= 0x01;
    NVIC->IP[30] = 0x45;
    NVIC->ISER[0] |= (1 << 30);
}

void TIM4_IRQHandler(void){
    if(TIM4->SR & 1 << 0){
        GPIOE->ODR ^= 1 << 5;
    }
    TIM4->SR &= ~(1 << 0);
}

int main(void){
    IERG3810_LED_Init();
    IERG3810_clock_tree_init();
    IERG3810_NVIC_SetPriorityGroup(5);
    // arr, psc
    IERG3810_TIM3_Init(4999,7199);
    IERG3810_TIM4_Init(1249,7199);    // 72M/7200/1250/2 = 4
}

```

```
while(1){  
    ;  
}  
}
```

Experiment 5.4

```
#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
#include "FONT.H"
#include "SevenSegments.h"
#include "CFONT.H"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
    u32 i;
    for(i = 0; i < count; i++);
}

void IERG3810_key2_ExtInit(){
    // EXTI-2
    RCC->APB2ENR |= 1 << 6;
    GPIOE ->CRL &= 0xFFFFF0FF;
    GPIOE ->CRL |= 0x00000800;
    GPIOE ->BSRR = 1 << 2;
    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[0] &= 0xFFFFF0FF;
    AFIO ->EXTICR[0] |= 0x00000400;
    EXTI ->IMR |= 1 << 2;
    EXTI ->FTSR |= 1 << 2;
    //EXTI ->RTSR |= 1 << 2;

    NVIC ->IP[8] = 0x65;
    NVIC ->ISER[0] &= ~(1 << 8);
    NVIC ->ISER[0] |= (1 << 8);
}

void IERG3810_NVIC_SetPriorityGroup(u8 prigroup){
    u32 temp, temp1;
    temp1 = prigroup & 0x00000007;
    temp1 <=< 8;
    temp = SCB ->AIRCR;
    temp &= 0x0000F8FF;
```

```

    temp |= 0x05FA0000;
    temp |= temp1;
    SCB -> AIRCR = temp;
}

void EXTI2_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOB -> BRR = 1 << 5;
        Delay(1000000);
        GPIOB -> BSRR = 1 << 5;
        Delay(1000000);
    }
    EXTI -> PR = 1 << 2;
}

void EXTI0_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOE -> BRR = 1 << 5; // on
        Delay(1000000);
        GPIOE -> BSRR = 1 << 5; // off
        Delay(1000000);
    }
    EXTI -> PR = 1 << 0;
}

void IERG3810_keyUP_ExtInit(){
    //PA0, KeyUp press = high, EXTI-0
    RCC->APB2ENR |= 1 << 2;
    GPIOA -> CRL &= 0xFFFFFFFF0;
    GPIOA -> CRL |= 0x00000008;
    GPIOA -> ODR |= 1 << 4;

    RCC->APB2ENR |= 0x1;
    AFIO -> EXTICR[0] &= 0xFFFFFFFF0; //EXTI-0
    AFIO -> EXTICR[0] |= 0x00000000; //EXTI-0, 0000: PA[x] pin

    EXTI -> IMR |= 0x1; //0001 not mask on line 0
    EXTI -> FTSR |= 0x1; // Falling trigger enabled

    NVIC -> IP[6] = 0x95; //priority = 0x95
    NVIC -> ISER[0] &= ~(1 << 6); //enable IRQ 6 for EXTI0
    NVIC -> ISER[0] |= (1 << 6); //IRQ 6
}

```



```

void IERG3810_PS2key_ExtInit(){
    // PS2 data : PC10, PS2 CLK: PC11

    RCC->APB2ENR |= 1 << 4;
    GPIOC ->CRH &= 0xFFFF00FF;//PC10, PC11
    GPIOC ->CRH |= 0x00008800; // 1000

    GPIOC ->BSRR = 1 << 11; // set high
    GPIOC ->BSRR = 1 << 10;

    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[2] &= 0xFFFF0FFF; //EXTI11
    AFIO ->EXTICR[2] |= 0x00002000;// 0010: PC11 pin

    EXTI ->IMR |= 1 << 11; // Event request from Line x is not masked
    EXTI ->FTSR |= 1 << 11;// Falling trigger enabled

    NVIC ->IP[40] = 0x65; //priority for IRQ 40
    NVIC ->ISER[1] |= (1 << 8); //enable IRQ 40 for EXTI[15:10]
}

u32 sheep = 0;
u32 timeout = 10000;
u32 ps2key = 0;
u32 tmp = 0;
u32 ps2count = 0;
u8 ps2dataReady = 0;
u8 key_stack[2];

void EXTI15_10_IRQHandler(void){

    if (ps2count > 0 && ps2count < 9){ //1:8 , bit 0:7 data

        tmp = ps2key >>= 1; //right shift 1
        if ((GPIOC->IDR)&(1<<10)){
            tmp |= 0x80;
        }
        ps2key = tmp;
        ps2count++;
    }

    else {
        ps2count++;
    }

    Delay(10);
}

```

```

    EXTI->PR = 1 << 11;
}

void IERG3810_TIM3_Init(u16 arr, u16 psc){

    //Tim3, IRQ#29
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;
    TIM3->PSC = psc;
    TIM3->DIER |= 1 << 0;
    TIM3->CR1 |= 0x01;
    NVIC->IP[29] = 0x45;
    NVIC->ISER[0] |= (1 << 29);
}

void DS0_turnOff(){
    GPIOB->BSRR = 1<<5;
}

void DS0_turnOff2(){
    DS0_turnOff();
}

void TIM3_IRQHandler(void){
    /*
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    TIM3->SR &= ~(1<<0);
    TIM3->SR &= ~(1<<0);
    */
    GPIOB->BSRR = 1 << 5;
    DS0_turnOff();
    DS0_turnOff2();
}

void IERG3810_TIM4_Init(u16 arr, u16 psc){

    //Tim4, IRQ#30

```

```

RCC->APB1ENR |= 1 << 2;    // TIM4
TIM4->ARR = arr;
TIM4->PSC = psc;
TIM4->DIER |= 1 << 0;
TIM4->CR1 |= 0x01;
NVIC->IP[30] = 0x45;
NVIC->ISER[0] |= (1 << 30);
}

void TIM4_IRQHandler(void){
    if(TIM4->SR & 1 << 0){
        GPIOE->ODR ^= 1 << 5;
    }
    TIM4->SR &=~(1 << 0);
}

int main(void){
    IERG3810_LED_Init();
    IERG3810_clock_tree_init();
    IERG3810_NVIC_SetPriorityGroup(5);
    // arr, psc
    IERG3810_TIM3_Init(4999,7199);
    IERG3810_TIM4_Init(1249,7199);    // 72M/7200/1250/2 = 4

    while(1){
        /*
        GPIOB->BSRR = 1 << 5;
        DS0_turnOff();
        DS0_turnOff2();
        */
        ;
    }
}

```

Experiment 5.5

```

#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
#include "FONT.H"

```

```

#include "SevenSegments.h"
#include "CFONT.H"
#include "global.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
    u32 i;
    for(i = 0; i < count; i++);
}

void IERG3810_key2_ExtInit(){
    // EXTI-2
    RCC->APB2ENR |= 1 << 6;
    GPIOE ->CRL &= 0xFFFFF0FF;
    GPIOE ->CRL |= 0x00000800;
    GPIOE ->BSRR = 1 << 2;
    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[0] &= 0xFFFFF0FF;
    AFIO ->EXTICR[0] |= 0x00000400;
    EXTI ->IMR |= 1 << 2;
    EXTI ->FTSR |= 1 << 2;
    //EXTI ->RTSR |= 1 << 2;

    NVIC ->IP[8] = 0x65;
    NVIC ->ISER[0] &= ~(1 << 8);
    NVIC ->ISER[0] |= (1 << 8);
}

void IERG3810_NVIC_SetPriorityGroup(u8 prigroup){
    u32 temp, temp1;
    temp1 = prigroup & 0x00000007;
    temp1 <<= 8;
    temp = SCB ->AIRCRR;
    temp &= 0x0000F8FF;
    temp |= 0x05FA0000;
    temp |= temp1;
    SCB ->AIRCRR = temp;
}

void EXTI2_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){

```

```

        GPIOB ->BRR = 1 << 5;
        Delay(1000000);
        GPIOB ->BSRR = 1 << 5;
        Delay(1000000);
    }
    EXTI ->PR = 1 << 2;
}

void EXTI0_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOE ->BRR = 1 << 5; // on
        Delay(1000000);
        GPIOE ->BSRR = 1 << 5; // off
        Delay(1000000);
    }
    EXTI ->PR = 1 << 0;
}

void IERG3810_keyUP_ExtInit(){
    //PA0, KeyUp press = high, EXTI-0
    RCC->APB2ENR |= 1 << 2;
    GPIOA ->CRL &= 0xFFFFFFFF0;
    GPIOA ->CRL |= 0x00000008;
    GPIOA ->ODR |= 1 << 4;

    RCC->APB2ENR |= 0x1;
    AFIO ->EXTICR[0] &= 0xFFFFFFFF0; //EXTI-0
    AFIO ->EXTICR[0] |= 0x00000000; //EXTI-0, 0000: PA[x] pin

    EXTI ->IMR |= 0x1; //0001 not mask on line 0
    EXTI ->FTSR |= 0x1; // Falling trigger enabled

    NVIC ->IP[6] = 0x95; //priority = 0x95
    NVIC ->ISER[0] &= ~(1 << 6); //enable IRQ 6 for EXTI0
    NVIC ->ISER[0] |= (1 << 6); //IRQ 6
}

void IERG3810_PS2key_ExtInit(){
    // PS2 data : PC10, PS2 CLK: PC11

    RCC->APB2ENR |= 1 << 4;
    GPIOC ->CRH &= 0xFFFF00FF; //PC10, PC11
    GPIOC ->CRH |= 0x00008800; // 1000

    GPIOC ->BSRR = 1 << 11; // set high

```

```

GPIOC ->BSRR = 1 << 10;

RCC->APB2ENR |= 0x01;
AFIO ->EXTICR[2] &= 0xFFFF0FFF; //EXTI11
AFIO ->EXTICR[2] |= 0x00002000; // 0010: PC11 pin

EXTI ->IMR |= 1 << 11; // Event request from Line x is not masked
EXTI ->FTSR |= 1 << 11; // Falling trigger enabled

NVIC ->IP[40] = 0x65; //priority for IRQ 40
NVIC ->ISER[1] |= (1 << 8); //enable IRQ 40 for EXTI[15:10]
}

u32 sheep = 0;
u32 timeout = 10000;
u32 ps2key = 0;
u32 tmp = 0;
u32 ps2count = 0;
u8 ps2dataReady = 0;
u8 key_stack[2];

void EXTI15_10_IRQHandler(void){

    if (ps2count > 0 && ps2count < 9){ //1:8 , bit 0:7 data

        tmp = ps2key >>= 1; //right shift 1
        if ((GPIOC->IDR)&(1<<10)){
            tmp |= 0x80;
        }
        ps2key = tmp;
        ps2count++;
    }

    else {
        ps2count++;
    }

    Delay(10);
    EXTI->PR = 1 << 11;
}

void IERG3810_TIM3_Init(u16 arr, u16 psc){

    //Tim3, IRQ#29
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;

```

```

    TIM3->PSC = psc;
    TIM3->DIER |= 1 << 0;
    TIM3->CR1 |= 0x01;
    NVIC->IP[29] = 0x45;
    NVIC->ISER[0] |= (1 << 29);
}

void TIM3_IRQHandler(void){
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    TIM3->SR &= ~(1<<0);
    TIM3->SR &= ~(1<<0);
}

void IERG3810_TIM4_Init(u16 arr, u16 psc){

    //Tim4, IRQ#30
    RCC->APB1ENR |= 1 << 2;      // TIM4
    TIM4->ARR = arr;
    TIM4->PSC = psc;
    TIM4->DIER |= 1 << 0;
    TIM4->CR1 |= 0x01;
    NVIC->IP[30] = 0x45;
    NVIC->ISER[0] |= (1 << 30);
}

void TIM4_IRQHandler(void){
    if(TIM4->SR & 1 << 0){
        GPIOE->ODR ^= 1 << 5;
    }
    TIM4->SR &= ~(1 << 0);
}

void DS0_turnOff(){
    GPIOB->BSRR = 1<<5;
}

void DS0_turnOff2(){

```

```

    DS0_turnOff();
}

void IERG3810_SYSTICK_Init10ms(void){

    SysTick->CTRL = 0;
    SysTick->LOAD = 0x00015F90;
    SysTick->CTRL |= 0x00000003;
}

int main(void){
    IERG3810_LED_Init();
    IERG3810_clock_tree_init();
    IERG3810_NVIC_SetPriorityGroup(5);
    // arr, psc
    IERG3810_TIM3_Init(4999,7199);
    IERG3810_TIM4_Init(1249,7199);    // 72M/7200/1250/2 = 4
    IERG3810_SYSTICK_Init10ms();

    // 100 HeartBeat = 1s
    // T1: 5 times in 1 second, 1 time is 1/5 second = 20 HeartBeat
    // T2: 3 times in 1 second, 1 time is 1/3 second = 100/3 = 33.333 HeartBeat
    while(1){
        if(task1HeartBeat >= 11){
            if (task1HeartBeat % 20 == 0){
                GPIOB->ODR ^= 1 << 5;

            }
            if (task1HeartBeat % 33 == 0){
                GPIOE->ODR ^= 1 << 5;

            }
        }
    }
}

```

Experiment 5.6

```

#include "stm32f10x.h"
#include "IERG3810_LED.h"
#include "IERG3810_Buzzer.h"
#include "IERG3810_KEY.h"
#include "IERG3810_USART.h"
#include "IERG3810_Clock.h"
#include "FONT.H"

```



```

#include "SevenSegments.h"
#include "CFONT.H"
#include "global.h"

void IERG3810_clock_tree_init(void);
void IERG3810_USART2_init(u32, u32);
void IERG3810_USART1_init(u32, u32);
void Delay(u32);
void USART_print(u8, char *);

void Delay(u32 count){
    u32 i;
    for(i = 0; i < count; i++);
}

void IERG3810_key2_ExtInit(){
    // EXTI-2
    RCC->APB2ENR |= 1 << 6;
    GPIOE ->CRL &= 0xFFFFF0FF;
    GPIOE ->CRL |= 0x00000800;
    GPIOE ->BSRR = 1 << 2;
    RCC->APB2ENR |= 0x01;
    AFIO ->EXTICR[0] &= 0xFFFFF0FF;
    AFIO ->EXTICR[0] |= 0x00000400;
    EXTI ->IMR |= 1 << 2;
    EXTI ->FTSR |= 1 << 2;
    //EXTI ->RTSR |= 1 << 2;

    NVIC ->IP[8] = 0x65;
    NVIC ->ISER[0] &= ~(1 << 8);
    NVIC ->ISER[0] |= (1 << 8);
}

void IERG3810_NVIC_SetPriorityGroup(u8 prigroup){
    u32 temp, temp1;
    temp1 = prigroup & 0x00000007;
    temp1 <<= 8;
    temp = SCB ->AIRCRR;
    temp &= 0x0000F8FF;
    temp |= 0x05FA0000;
    temp |= temp1;
    SCB ->AIRCRR = temp;
}

void EXTI2_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){

```

```

        GPIOB ->BRR = 1 << 5;
        Delay(1000000);
        GPIOB ->BSRR = 1 << 5;
        Delay(1000000);
    }
    EXTI ->PR = 1 << 2;
}

void EXTI0_IRQHandler(void){
    u8 i;
    for(i = 0; i < 10; i++){
        GPIOE ->BRR = 1 << 5; // on
        Delay(1000000);
        GPIOE ->BSRR = 1 << 5; // off
        Delay(1000000);
    }
    EXTI ->PR = 1 << 0;
}

void IERG3810_keyUP_ExtInit(){
    //PA0, KeyUp press = high, EXTI-0
    RCC->APB2ENR |= 1 << 2;
    GPIOA ->CRL &= 0xFFFFFFFF0;
    GPIOA ->CRL |= 0x00000008;
    GPIOA ->ODR |= 1 << 4;

    RCC->APB2ENR |= 0x1;
    AFIO ->EXTICR[0] &= 0xFFFFFFFF0; //EXTI-0
    AFIO ->EXTICR[0] |= 0x00000000; //EXTI-0, 0000: PA[x] pin

    EXTI ->IMR |= 0x1; //0001 not mask on line 0
    EXTI ->FTSR |= 0x1; // Falling trigger enabled

    NVIC ->IP[6] = 0x95; //priority = 0x95
    NVIC ->ISER[0] &= ~(1 << 6); //enable IRQ 6 for EXTI0
    NVIC ->ISER[0] |= (1 << 6); //IRQ 6
}

void IERG3810_PS2key_ExtInit(){
    // PS2 data : PC10, PS2 CLK: PC11

    RCC->APB2ENR |= 1 << 4;
    GPIOC ->CRH &= 0xFFFF00FF; //PC10, PC11
    GPIOC ->CRH |= 0x00008800; // 1000

    GPIOC ->BSRR = 1 << 11; // set high

```

```

GPIOC ->BSRR = 1 << 10;

RCC->APB2ENR |= 0x01;
AFIO ->EXTICR[2] &= 0xFFFF0FFF; //EXTI11
AFIO ->EXTICR[2] |= 0x00002000; // 0010: PC11 pin

EXTI ->IMR |= 1 << 11; // Event request from Line x is not masked
EXTI ->FTSR |= 1 << 11; // Falling trigger enabled

NVIC ->IP[40] = 0x65; //priority for IRQ 40
NVIC ->ISER[1] |= (1 << 8); //enable IRQ 40 for EXTI[15:10]
}

u32 sheep = 0;
u32 timeout = 10000;
u32 ps2key = 0;
u32 tmp = 0;
u32 ps2count = 0;
u8 ps2dataReady = 0;
u8 key_stack[2];

void EXTI15_10_IRQHandler(void){

    if (ps2count > 0 && ps2count < 9){ //1:8 , bit 0:7 data

        tmp = ps2key >>= 1; //right shift 1
        if ((GPIOC->IDR)&(1<<10)){
            tmp |= 0x80;
        }
        ps2key = tmp;
        ps2count++;
    }

    else {
        ps2count++;
    }

    Delay(10);
    EXTI->PR = 1 << 11;
}

void IERG3810_TIM3_Init(u16 arr, u16 psc){

    //Tim3, IRQ#29
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;

```

```

    TIM3->PSC = psc;
    TIM3->DIER |= 1 << 0;
    TIM3->CR1 |= 0x01;
    NVIC->IP[29] = 0x45;
    NVIC->ISER[0] |= (1 << 29);
}

void TIM3_IRQHandler(void){
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->BRR = 1 << 5;
    GPIOB->BSRR = 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR ^= 1 << 5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    GPIOB->ODR &= ~(1<<5);
    GPIOB->ODR |= 1<<5;
    TIM3->SR &= ~(1<<0);
    TIM3->SR &= ~(1<<0);
}

void IERG3810_TIM4_Init(u16 arr, u16 psc){

    //Tim4, IRQ#30
    RCC->APB1ENR |= 1 << 2;      // TIM4
    TIM4->ARR = arr;
    TIM4->PSC = psc;
    TIM4->DIER |= 1 << 0;
    TIM4->CR1 |= 0x01;
    NVIC->IP[30] = 0x45;
    NVIC->ISER[0] |= (1 << 30);
}

void TIM4_IRQHandler(void){
    if(TIM4->SR & 1 << 0){
        GPIOE->ODR ^= 1 << 5;
    }
    TIM4->SR &= ~(1 << 0);
}

void DS0_turnOff(){
    GPIOB->BSRR = 1<<5;
}

void DS0_turnOff2(){

```

```

    DS0_turnOff();
}

void IERG3810_SYSTICK_Init10ms(void){

    SysTick->CTRL = 0;
    SysTick->LOAD = 0x00015F90;
    SysTick->CTRL |= 0x00000003;
}

void IERG3810_TIM3_PwmInit(u16 arr, u16 psc){
    RCC->APB2ENR |= 1 << 3;
    GPIOB ->CRL &= 0xFF0FFFFFFF;
    GPIOB ->CRL |= 0x00B00000;
    RCC->APB2ENR |= 1 << 0;
    AFIO->MAPR &= 0xFFFFF3FF;
    AFIO->MAPR |= 1 << 11;
    RCC->APB1ENR |= 1 << 1;
    TIM3->ARR = arr;
    TIM3->PSC = psc;
    TIM3->CCMR1 |= 7 << 12;
    TIM3->CCMR1 |= 1 << 11;
    TIM3->CCER |= 1 << 4;
    TIM3->CR1 = 0x0080;
    TIM3->CR1 |= 0x01;
}

int main(void){
    u16 led0pwnval = 0;
    u8 dir = 1;
    IERG3810_clock_tree_init();
    IERG3810_LED_Init();
    IERG3810_TIM3_PwmInit(6666, 72);
    //150

    // 100 HeartBeat = 1s
    // T1: 5 times in 1 second, 1 time is 1/5 second = 20 HeartBeat
    // T2: 3 times in 1 second, 1 time is 1/3 second = 100/3 = 33.333 HeartBeat
    while(1){
        Delay(1500);
        if(dir){
            led0pwnval++;
        }
        else{
            led0pwnval--;
        }
        if(led0pwnval > 5000){
            dir = 0;

```

```
    }
    if(led0pwnval == 0){
        dir = 1;
    }
    TIM3->CCR2 = led0pwnval;
}
}
```

III. DISCUSSION

Exp 5.1

<Question: Explain the programs of figure 5-2 in your lab report.>

1. main():

The clock signal is divided by the pre-scalar and counter. The LED flashing rate is equal to $72000000\text{Hz} / 7200 / 5000 / 2 = 1\text{Hz}$.

2. TIM3_Init():

Configuration of the arr and psc for TIM3, also enable interrupt and setting it's priority level.

3. TIM3_IRQHandler():

Check the uif, toggle the ds0 and clear UIF.

Exp 5.2

<Question: Compare the results for DS0.>

DS0 is flashing in 1Hz, DS1 with 4Hz flashing rate and control by timer-4. DS1 will flash faster than DS0.

Exp 5.3

< Question: Modify the codes to show the difference between read modify write and direct modify register>

Different approach have different overhead problem and running time.

Exp 5.4

< Question: Discuss your observed results>

The two approaches have different overhead problem, should be using an oscilloscope to detect it.

Exp 5.5

<Question: Discuss the technique of multi-tasking program with 10ms SysTick and how to apply it into your mini-project.>

Using the SysTick can allowing multi-tasking by tracking the heartbeat. 10ms can perform real time operation, like checking the status in the game.

IV. SUMMARY

We learn the Timer setting of Cortex-M3.

V. DIVISION OF WORK

<Lab work: Jesse & Derek, Report writing: Jesse>

VI. REFERENCES