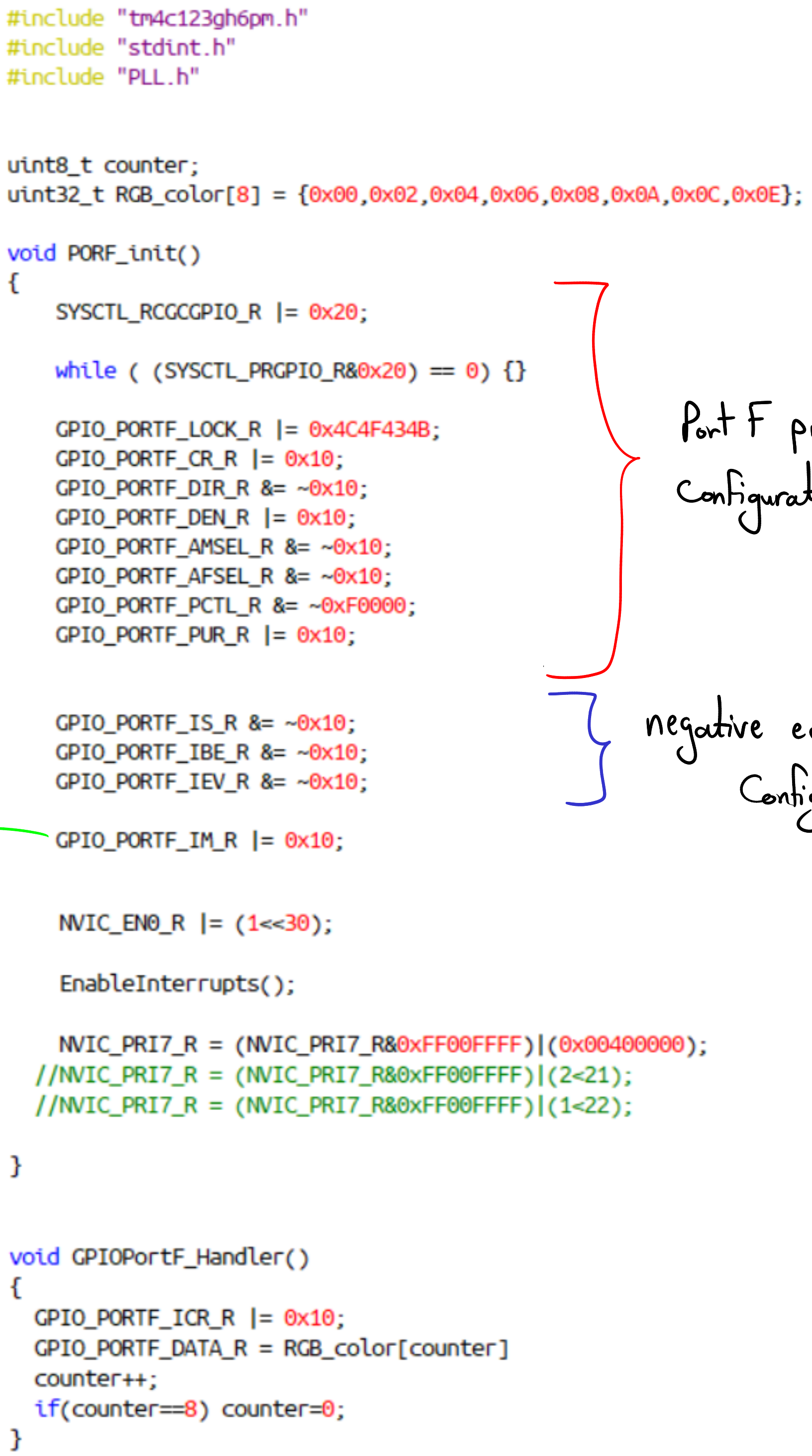
Faculty of Engineering
Computer and Systems Engineering Department

CSE 211 [Spring 2022]
Introduction to Embedded Systems

SHEET 8

Q1. Write using C, a function to initialize port F pin 4 as digital input with negative edge triggered Interrupt with priority 2 and write the ISR which changes the color of RGB in a cyclic way from 000 to 111 then to 000 again.



```c
#include "tm4c123gh6pm.h"
#include "stdint.h"
#include "PLL.h"


uint8_t counter;
uint32_t RGB_color[8] = {0x00,0x02,0x04,0x06,0x08,0x0A,0x0C,0x0E};

void PORF_init()
{
    SYSCTL_RCGCGPIO_R  |= 0x20;

    while ( (SYSCTL_PRGPIO_R&0x20) == 0) {}

    GPIO_PORTF_LOCK_R  |= 0x4C4F434B;
    GPIO_PORTF_CR_R  |= 0x10;
    GPIO_PORTF_DIR_R  &= ~0x10;
    GPIO_PORTF_DEN_R  |= 0x10;
    GPIO_PORTF_AMSEL_R  &= ~0x10;
    GPIO_PORTF_AFSEL_R  &= ~0x10;
    GPIO_PORTF_PCTL_R  &= ~0xF0000;
    GPIO_PORTF_PUR_R  |= 0x10;


    GPIO_PORTF_IS_R  &= ~0x10;
    GPIO_PORTF_IBE_R  &= ~0x10;
    GPIO_PORTF_IEV_R  &= ~0x10;

    GPIO_PORTF_IM_R  |= 0x10;


    NVIC_EN0_R  |= (1<<30);

    EnableInterrupts();

    NVIC_PRI7_R = (NVIC_PRI7_R&0xFF00FFFF)|(0x00400000);
  //NVIC_PRI7_R = (NVIC_PRI7_R&0xFF00FFFF)|(2<21);
  //NVIC_PRI7_R = (NVIC_PRI7_R&0xFF00FFFF)|(1<22);

}


void GPIOPortF_Handler()
{
  GPIO_PORTF_ICR_R  |= 0x10;
  GPIO_PORTF_DATA_R = RGB_color[counter]
  counter++;
  if(counter==8) counter=0;
}
```

Port F pin 4
Configuration as digital input

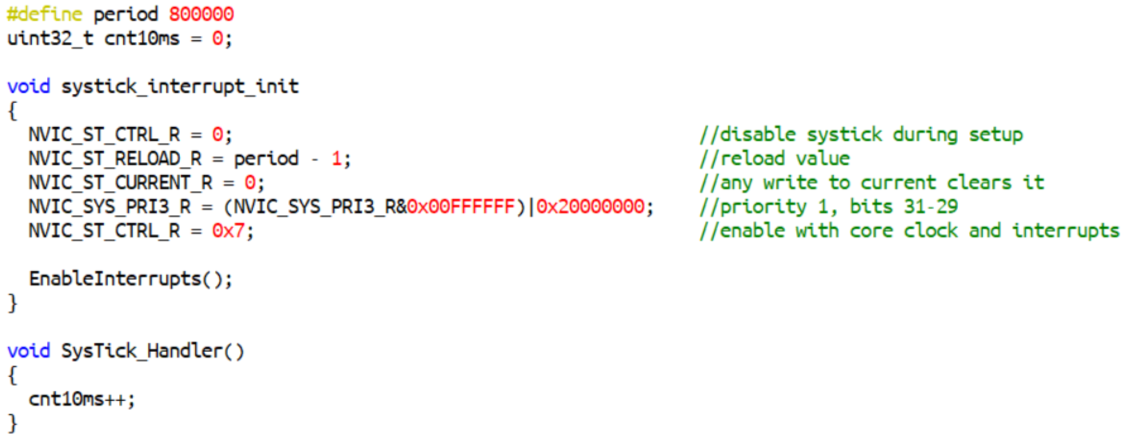
negative edge triggered
Configuration

arm interrupt
port F pin 4

Method -

Q2. Write using C, a function to initialize SysTick periodic interrupt each 10 ms with priority 1 (assume *system* clock is 80 MHz) and write an ISR which increments a global variable "cnt10ms" by

① period = 80MHz * 10msec = 800,000



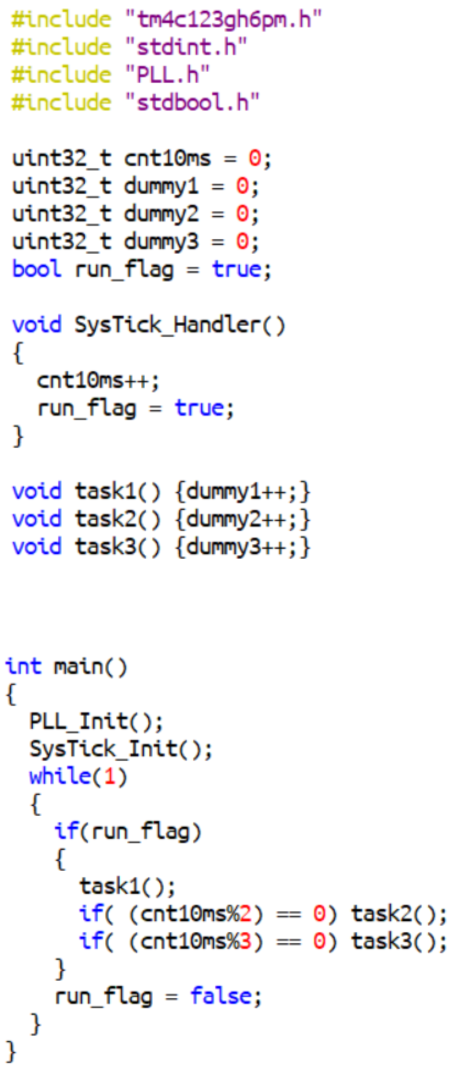
```c
#define period 800000
uint32_t cnt10ms = 0;

void systick_interrupt_init
{
  NVIC_ST_CTRL_R = 0;                                          //disable systick during setup
  NVIC_ST_RELOAD_R = period - 1;                               //reload value
  NVIC_ST_CURRENT_R = 0;                                       //any write to current clears it
  NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0x20000000;   //priority 1, bits 31-29
  NVIC_ST_CTRL_R = 0x7;                                        //enable with core clock and interrupts

  EnableInterrupts();
}

void SysTick_Handler()
{
  cnt10ms++;
}
```

Q3. Write using C, a main function that calls 3 functions which are task1(), task2(), and task3(). Task1 should run every 10 ms, task 2 should run every 20 ms, and task 3 should run every 30 ms. Assume there is a global variable "cnt10ms" that is initialized by 0 and is incremented by 1 every 10 ms.



```c
#include "tm4c123gh6pm.h"
#include "stdint.h"
#include "PLL.h"
#include "stdbool.h"

uint32_t cnt10ms = 0;
uint32_t dummy1 = 0;
uint32_t dummy2 = 0;
uint32_t dummy3 = 0;
bool run_flag = true;

void SysTick_Handler()
{
  cnt10ms++;
  run_flag = true;
}

void task1() {dummy1++;}
void task2() {dummy2++;}
void task3() {dummy3++;}


int main()
{
  PLL_Init();
  SysTick_Init();
  while(1)
  {
    if(run_flag)
    {
      task1();
      if( (cnt10ms%2) == 0) task2();
      if( (cnt10ms%3) == 0) task3();
    }
    run_flag = false;
  }
}
```