

КАФЕДРА №

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

Настройка системы прерываний

Микроконтроллера

по курсу: Программирование встроенных приложений

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2024

## 1. Задание

| Вариант | Номера<br>линий PC | Количество групп<br>и подгрупп<br>приоритетов |          | Номера приоритетов до настройки<br>группировки приоритетов и после,<br>соответственно порядку линий |                              | Приоритет<br>SysTick |
|---------|--------------------|---|----------|---|------------------------------|----------------------|
|         |                    | Групп   | Подгрупп | до  | после<br>№группы(№подгруппы) |                      |
| 24      | 0,1,4,5            | 2   | 8        | 55,70,84,69   | 0(3), 1(3), 0(2), 1(2)       | 6                    |

## 2. Листинг

```
#include "RTE_Components.h" // Component selection
#include CMSIS_device_header // Device header
#include <stdio.h>

static volatile uint32_t ui_count100ms=0;
void delay(void){
    volatile uint32_t i=6000000;
    while(i > 0)
        i--;
}

int main(void) {
    uint32_t priGroup = 0, PreemptPriority=0, SubPriority=0;
    //настройка частоты 72 МГц
    SET_BIT(RCC -> CR,RCC_CR_HSEON);
    while((RCC->CR & RCC_CR_HSERDY)==0){ }
    FLASH->ACR = FLASH_ACR_PRFTBE|FLASH_ACR_LATENCY_1;
    RCC->CFGR |= (uint32_t)(RCC_CFGR_PLLSRC_HSE_PREDIV | RCC_CFGR_PLLMUL9);
    SET_BIT(RCC -> CR,RCC_CR_PLLON);
    while((RCC->CR & RCC_CR_PLLRDY) == 0){ }
    RCC->CFGR &= (uint32_t)((uint32_t)~(RCC_CFGR_SW));
    RCC->CFGR |= (uint32_t)RCC_CFGR_SW_PLL;
    while ((RCC->CFGR & (uint32_t)RCC_CFGR_SWS) != (uint32_t)RCC_CFGR_SWS_PLL){ }
    SystemCoreClockUpdate();//проверяем частоту SystemCoreClock
    printf("clk=%d\n",SystemCoreClock);
    SET_BIT(RCC -> APB2ENR, RCC_APB2ENR_SYSCFGEN);//разрешаем тактирование
    SYSCFG
```

```

SET_BIT(RCC -> AHBENR, RCC_AHBENR_GPIOCEN); //GPIOC
CLEAR_BIT(GPIOC->MODER,GPIO_MODER_MODER0|GPIO_MODER_MODER1|
GPIO_MODER_MODER4|GPIO_MODER_MODER5); //PC2,3,4,6 In
SET_BIT(GPIOC->PUPDR,GPIO_PUPDR_PUPDR0_0|GPIO_PUPDR_PUPDR1_0|
GPIO_PUPDR_PUPDR4_0|GPIO_PUPDR_PUPDR5_0); //Pull up PC2,3,4,6
SET_BIT(GPIOC->MODER,GPIO_MODER_MODER2_0|GPIO_MODER_MODER3_0|
GPIO_MODER_MODER6_0|GPIO_MODER_MODER7_0); //PC0,1,5,7 Out
SET_BIT(GPIOC->OTYPER, GPIO_OTYPER_OT_2|GPIO_OTYPER_OT_3|
GPIO_OTYPER_OT_6|GPIO_OTYPER_OT_7); //режим с открытым стоком
SET_BIT(GPIOC-
>BRR,GPIO_BRR_BR_2|GPIO_BRR_BR_3|GPIO_BRR_BR_6|GPIO_BRR_BR_7);
//притягиваем к нулю
NVIC_SetPriorityGrouping(2);
priGroup = NVIC_GetPriorityGrouping();
printf("Priority Group=%d\r\n",priGroup);
NVIC_SetPriority(EXTI0_IRQn,3);

NVIC_DecodePriority(NVIC_GetPriority(EXTI0_IRQn),priGroup,&PreemptPriority,&SubPriority
);
printf("EXTI0 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);
NVIC_SetPriority(EXTI1_IRQn,3);

NVIC_DecodePriority(NVIC_GetPriority(EXTI1_IRQn),priGroup,&PreemptPriority,&SubPriority
);
printf("EXTI1 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);
NVIC_SetPriority(EXTI4_IRQn,2);

NVIC_DecodePriority(NVIC_GetPriority(EXTI4_IRQn),priGroup,&PreemptPriority,&SubPriority
);
printf("EXTI4 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);
NVIC_SetPriority(EXTI9_5_IRQn,2);

NVIC_DecodePriority(NVIC_GetPriority(EXTI9_5_IRQn),priGroup,&PreemptPriority,&SubPrior
ity);
printf("EXTI5 Preempt Priority=%d \tSubPriority=%d\r\n",PreemptPriority,SubPriority);
printf("Press any key\r\n");
//прерывание на спад сигнала
SET_BIT(EXTI-
>FTSR,EXTI_FTSR_FT0|EXTI_FTSR_FT1|EXTI_FTSR_FT4|EXTI_FTSR_FT5);
//разрешаем прерывания внешних линий 2,3,4,6

```

```

SET_BIT(EXTI->IMR,EXTI_IMR_IM0|EXTI_IMR_IM1|EXTI_IMR_IM4|EXTI_IMR_IM5);
// выбираем в качестве внешних входов EXTI линии:
//EXTI2=PC2 EXTI3=PC3 EXTI4=PC4 EXTI6=PC6
SYSCFG->EXTICR[0]=SYSCFG_EXTICR1_EXTI0_PC|SYSCFG_EXTICR1_EXTI1_PC;
SYSCFG->EXTICR[1]=SYSCFG_EXTICR2_EXTI4_PC|SYSCFG_EXTICR2_EXTI5_PC;
NVIC_EnableIRQ(EXTI0_IRQn);
NVIC_EnableIRQ(EXTI1_IRQn);
NVIC_EnableIRQ(EXTI4_IRQn);
NVIC_EnableIRQ(EXTI9_5_IRQn);
SysTick_Config(0x6DDD00);//прерывание каждые 100мсек
NVIC_SetPriority(SysTick_IRQn,8);
while(1){ }
}

void SysTick_Handler(void){
    ui_count100ms++;
    if(ui_count100ms%3==0)//выводим каждые 0,3 секунды
        ITM_SendChar('o');
}

void EXTI0_IRQHandler(void){
    EXTI->PR = EXTI_PR_PR0;
    ITM_SendChar('0');
    delay();
    ITM_SendChar('a');
    ITM_SendChar('\n');
}

void EXTI1_IRQHandler(void){
    EXTI->PR = EXTI_PR_PR1;
    ITM_SendChar('1');
    delay();
    ITM_SendChar('b');
    ITM_SendChar('\n');
}

void EXTI4_IRQHandler(void) {
    EXTI->PR = EXTI_PR_PR4;

```

```

    ITM_SendChar('4');
    delay();
    ITM_SendChar('c');
    ITM_SendChar('\n');
}

void EXTI9_5_IRQHandler(void){
    EXTI->PR = EXTI_PR_PR6;
    ITM_SendChar('5');
    delay();
    ITM_SendChar('d');
    ITM_SendChar('\n');
}

```

Debug Viewer:

